

# Estabilització de seqüències de vídeo mitjançant tècniques de processament d'imatge

6 de setembre de 2010

# Agraïments

Fa molt temps que vaig començar a treballar amb el meu projecte final de carrera i han estat moltes les coses que han anat passant durant aquest temps. No en va, vaig començar el meu primer projecte. L'experiència em va ensenyar que a vegades s'ha de ser persistent i buscar el camí encara que el guia no estigui per ensenyar-te'l. Aquell primer projecte, però, va acabar en ben poca cosa. El camí feia un revolt però continuava. Va ser llavors quan vaig aterrar al despatx de Javier, el que ha estat el meu director de projecte i qui em va començar a guiar fins el dia d'avui.

Han estat moltes les persones que han estat al meu costat durant aquest temps i a qui he d'agrair el seu ajut. Però més que a ningú he d'agrair el suport dels meus pares. Ells sempre m'han animat a que continués treballant i el pogués enllestir. Tot i això mai m'han posat cap pressió i els hi estic molt agraït. També la meva germana amb les seves amenaces que aconseguiria el títol abans que jo m'ha animat a continuar amb el treball. També he de donar les gràcies a Andreu, qui m'ha acompanyat durant totes les fases per les que he passat. M'ha donat suport moral i tècnic i m'ha ajudat en alguns moments a superar algun revolt. I a Ilaria, qui m'ha posat pressió quan la necessitava, quan ja gairebé ho tenia enllestit i em feia falta l'empenta final.

Durant els dos últims anys i mig he estat treballant mentre intentava fer el projecte final de carrera. És un poc difícil compaginar la feina de consultor i la realització del projecte i segurament per això he estat tant temps per poder-lo finalitzar. Malgrat això dono les gràcies als dos caps que he tingut, Joan i Gonzalo, per haver-me donat el seu consentiment per reduir la jornada en una ocasió i per agafar-me una excedència en una altra.

Precisament abans de començar l'excedència em van proposar de tornar a donar un gir en el camí. A començar un nou projecte dintre de l'empresa i que em serviria no tan sols per acabar la carrera sinó també per millorar la meva posició dintre de l'empresa. Vaig valorar l'opció amb deteniment i fins i tot vaig començar a treballar amb ella. Això però, significava renunciar al treball que feia temps havia començat per segon cop. Una xerrada amb un altre professor, Josep M., em va ajudar a acabar de decidir que aquest no era el camí quan em va dir: *“A tothom li passa que en algun moment s'entrebanca i ho deixaria tot de banda, llavors és quan més èmfasi s'ha de posar per poder acabar el que has començat.”*

Finalment he de donar les gràcies a tothom qui ha estat al meu costat durant aquest temps i ha aguantat les meves historietes, a vegades poc interessants, sobre el que estava fent. Júlia, que en va d'haver d'aguantar unes quantes i veia com cada vegada la cosa es feia més i més llarga. Dani, que sempre ha estat al meu costat escoltant pacientment tot el que li contava. Albert, qui m'ha donat suport tècnic amb el software i m'ha ajudat a fer algunes de les seqüències de prova. Sergi, qui ha desenvolupat un software per controlar les videocàmeres del laboratori del TSC per poder enregistrar les seqüències de prova. Fernando i Jaume, amb qui he compartit pis i també n'han sentit unes quantes. I com no, els companys de feina: Alberto, Lucas, Sergi, Daisy, Rafa, Lorenz, Javi, Quim, Isa i tants altres que sempre em preguntaven pel projecte i volien que els expliqués amb detall la feina que estava fent.

# Índex

<b>1</b>	<b>Motivació</b>	<b>4</b>
1.1	Introducció . . . . .	4
1.2	Objectius . . . . .	5
1.2.1	Eliminació de vibracions en seqüències en moviment . . . . .	5
1.2.2	Augment de la velocitat de processat . . . . .	6
1.3	Diagrama de blocs de l'aplicació . . . . .	7
<b>2</b>	<b>Estat de l'art i treball previ</b>	<b>9</b>
2.1	Estat de l'art . . . . .	9
2.2	Treball previ . . . . .	12
<b>3</b>	<b>Detecció de punts d'interès</b>	<b>15</b>
3.1	El detector de Harris . . . . .	16
3.2	El detector de Trajkovic-Headley . . . . .	19
3.3	Detector desenvolupat . . . . .	22
<b>4</b>	<b>Obtenció dels paràmetres de moviment</b>	<b>26</b>
4.1	Models de transformació . . . . .	27
4.2	Transformació translacional . . . . .	27
4.2.1	Transformació afí . . . . .	28
4.2.2	Transformació Perspectiva . . . . .	30
4.2.3	Elecció del model de transformació . . . . .	31
4.3	Estimació dels paràmetres de moviment . . . . .	33
4.3.1	Algoritme de Gauss-Newton . . . . .	34
4.3.2	Correlació . . . . .	36
	Resolució del model de transformació . . . . .	38
<b>5</b>	<b>Eliminació de vibracions i compensació</b>	<b>41</b>
5.1	Càmera sense moviment . . . . .	42
5.2	Càmera amb moviment conegut . . . . .	42
5.3	Càmera amb moviment desconegut a priori . . . . .	44

<i>ÍNDIX</i>	3
<b>6 Resultats</b>	<b>47</b>
6.1 Seqüència de vídeo amb la càmera estàtica . . . . .	47
6.1.1 Compensació de vibracions amb descens de gradient . . . . .	48
6.1.2 Compensació de vibracions mitjançant la correlació . . . . .	48
6.2 Seqüència amb la càmera amb moviment . . . . .	54
Estimació pel mètode de Gauss-Newton . . . . .	55
Estimació pel mètode de correlació . . . . .	57
6.2.1 Compensació amb el moviment conegut . . . . .	60
6.2.2 Compensació amb moviment desconegut . . . . .	61
<b>7 Conclusions i treball futur</b>	<b>66</b>
7.1 Conclusions . . . . .	66
7.2 Treball futur . . . . .	68



# Capítol 1

## Motivació

### 1.1 Introducció

El terme tecnologia existeix des de fa molt temps. Tot i això, no ha estat fins a mitjan segle XX que ha evolucionat per entendre-la tal com la coneixem avui en dia. Amb l'aparició i l'evolució del transistor la tecnologia ha experimentat un creixement grandíssim. A més a més tal com va evolucionant la tecnologia apareixen noves línies d'investigació que fins aleshores eren impensables. Precisament, això és el que ha succeït en el camp de processament de la imatge. Les aplicacions de processament d'imatge requereixen un gran cost computacional que no ha estat possible obtenir fins els últims anys gràcies a l'avenç en la miniaturització dels transistors integrats en plaques de silici. A la vegada, aquest avenç en les tècniques de processament d'imatge han permès el desenvolupament de noves aplicacions.

Fixem-nos, per exemple, en el cas de la seguretat. Fins fa poc, els únics elements de seguretat dels que es disposava eren els guardes, o més recentment, les alarmes. Actualment, aquesta situació ha canviat de forma dràstica gràcies als sistemes de videovigilància. A l'inici, aquests només permetien fer gravacions de les imatges capturades mitjançant les càmeres. Avui en dia, les seves funcionalitats han anat augmentant de forma espectacular. Així, ja podem trobar sistemes comercials que permeten detectar i seguir objectes en moviment, activar alarmes en cas que algú accedeixi a una zona restringida, llegir matrícules o fins i tot reconèixer cares de persones. Totes aquestes aplicacions requereixen la captura prèvia o en temps real de la seqüència de vídeo de la millor forma possible. Per tant, es fan necessaris els mètodes per captar el vídeo lliure de moviments indesitjats. Actualment existeixen solucions mecàniques per a aquest problema. Utilitzant petits servomotors es mou la lent que capta la llum de la càmera per tal de compensar el moviment extern. El gran avantatge d'aquests solucions és que capturen el vídeo ja lliure dels moviments indesitjats. Per altra banda, aquestes solucions són molt cares i incrementen molt el cost total del sistema ja que requereixen càmeres especials. El sistema de compensació de la vibració ha d'estar en totes les càmeres amb el cost de manteniment associat que comporta.

En el departament de Teoria del Senyal i Comunicacions (d'aquí endavant TSC) s'està duent a terme una línia d'investigació per compensar les vibracions mitjançant una solució software. Aquesta alternativa a la compensació mecànica és molt més econòmica ja que permet continuar utilitzant les càmeres convencionals sense sistemes addicionals. Aquest projecte s'emmarca dintre d'aquesta línia d'investigació i pretén estudiar la viabilitat d'aquesta solució.

Per tal de començar a desenvolupar el software d'estabilització de seqüències de vídeo, és necessari tenir alguns coneixements previs sobre tècniques utilitzades en processat d'imatge. Entre elles es poden destacar la cerca de punts d'interès en una imatge i la cerca de correspondències

entre dos frames d'una seqüència de vídeo. Durant l'execució d'aquest treball s'estudiaran algunes tècniques de cerca de punts d'interès documentades en la literatura i es buscaran noves solucions més adequades per al cas que ens ocupa. També es realitzarà un estudi comparatiu entre diversos tipus de cerca de correspondències per tal d'establir quina és la que proporciona millors resultats.

## 1.2 Objectius

Tal com ja s'ha comentat en el punt anterior, en el grup del TSC s'està duent a terme una línia d'investigació en el processament de la imatge i en concret en la compensació de vibracions en seqüències de vídeo procedents de càmeres de videovigilància. Partint del software disponible de processament de seqüències de vídeo, en aquest projecte és pretenen assolir dos objectius clarament diferenciats.

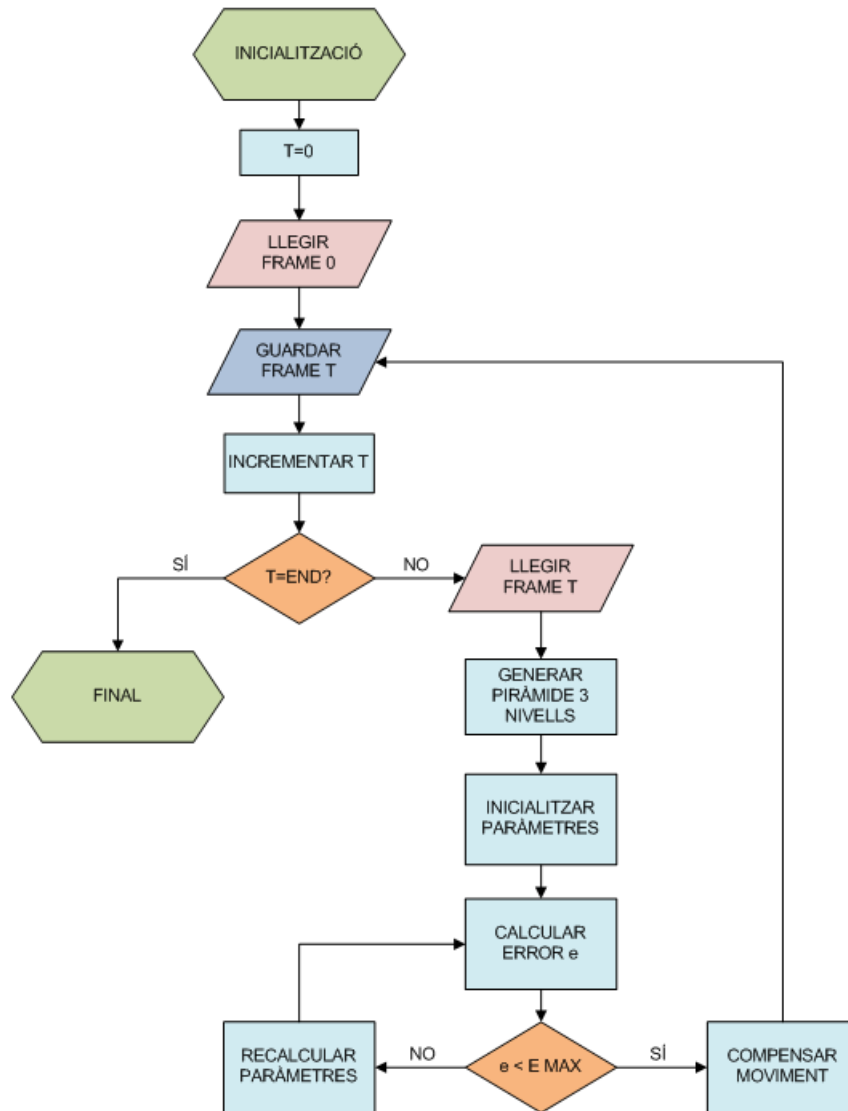
El primer d'ells consisteix en ampliar el software desenvolupat en el TSC. Aquest software realitza la compensació de vibracions de càmeres estàtiques. En aquest projecte s'utilitzarà aquest software com a base per poder compensar les vibracions de seqüències de vídeo de càmeres en moviment. El moviment de la càmera és necessari quan la zona a cobrir és major que el camp visual d'aquesta. Per tant, l'aplicació haurà d'estimar el moviment i distingir entre el de la càmera i les vibracions indesitjades. Un cop estimat el moviment total haurà de ser capaç d'eliminar les vibracions mantenint el moviment original de la càmera. S'estudiaran dos casos, el primer d'ells considerarà conegut el moviment de la càmera. En el segon cas el moviment de la càmera serà desconegut a priori.

El segon objectiu que es pretén assolir és la implementació d'un algoritme d'estimació de moviment més ràpid per tal de poder augmentar la velocitat de processat fins a 25fps ja que la nostra aplicació es basarà en l'estàndard de vídeo PAL. El software disponible realitza el processament a una velocitat de 0,5 fps. Malgrat que amb el temps i gràcies a l'augment de velocitat dels processadors es podria arribar a assolir la velocitat desitjada, en aquest projecte s'estudiaran alternatives de l'estimació de moviment per tal de poder millorar aquesta velocitat de processament. Per tal de poder compensar les vibracions a temps real serà necessari processar les seqüències a 25fps ja que aquesta és la velocitat de captura en l'estàndard PAL.

### 1.2.1 Eliminació de vibracions en seqüències en moviment

El programa actual realitza una estimació dels paràmetres de moviment entre el frame inicial de la seqüència ( $t_0$ ) i el frame actual ( $t_i$ ) per a la eliminació del moviment. En absència de moviment aquesta estimació hauria de trobar els paràmetres d'una càmera fixa. Si els paràmetres difereixen el programa realitza el compensat del frame actual per tal que el moviment final entre  $t_0$  i  $t_i$  sigui nul. L'algoritme 1.1 mostra el funcionament d'aquest programa. En primer lloc s'inicialitzen les variables necessàries i es llegeix el frame inicial que s'emmagatzema com a referència. En cada iteració el programa incrementa la variable T i llegeix el frame corresponent si existeix, sinó acaba. Amb el fotograma T i el de referència realitza l'estimació de moviment mitjançant un mètode de descens de gradient multi-nivell. Aquest mètode es descriu en més detall en la secció 2.2 a la pàgina 12 i 4.3.1 a la pàgina 34. Finalment es compensa el moviment i es procedeix a realitzar la següent iteració.

Per tal d'eliminar les vibracions en seqüències de vídeo en que la càmera pot tenir moviment, es fa necessari separar el moviment de la càmera de les vibracions. Existeixen dues situacions a tenir en compte:

**Algorithm 1.1** Compensació de moviment per a seqüències estàtiques

- Càmera amb moviment conegut a priori. Un cop estimat el moviment de la càmera el podem comparar amb el moviment conegut que hauria de realitzar. Si difereix haurem de compensar la diferència de moviment.
- Càmera amb moviment desconegut. En aquest cas la càmera realitza un moviment però no coneixem els seus paràmetres. Per tal de poder eliminar el moviment haurem de separar el moviment de la càmera de les vibracions. Per fer-ho s'haurà d'establir una restricció: el moviment degut a les vibracions tindrà components freqüencials més elevats que el moviment de la càmera. D'aquesta forma es podrà calcular el moviment de la càmera mitjançant un filtrat dels paràmetres de moviment estimats.

### 1.2.2 Augment de la velocitat de processat

El segon objectiu del projecte consisteix en realitzar el processat d'eliminació de vibracions en temps real. El software previ d'eliminació de vibracions per a seqüències de vídeo estàtiques

estima els paràmetres de moviment mitjançant la tècnica de descens de gradient multi-nivell explicada en 2.2 a la pàgina 12. El resultat d'aplicar aquest mètode proporciona uns resultats molt bons ja que els paràmetres de moviment són estimats amb molta precisió. El principal problema que presenta aquesta tècnica és que la velocitat de processament és molt baixa, al voltant de 0,5 frames per segon.

En aquest projecte es pretén modificar la tècnica d'estimació dels paràmetres per altres més eficients com podrien ser la correlació partint de certs punts d'interès dintre les imatges. En primer lloc haurem de buscar els punts i seguidament trobar el màxim de la correlació de les regions al voltant d'aquests. Un cop establertes les correspondències entre els frames, resoldrem el model de moviment utilitzat per tal d'obtenir els paràmetres corresponents.

L'objectiu marcat consisteix a augmentar entre 25 i 50 cops la velocitat de processat. Actualment el software compensa el moviment de seqüències estàtiques a 0.5 fps mentre que l'objectiu consisteix en assolir una velocitat màxima de 25 fps. Tot i això seria acceptable aconseguir velocitats al voltant de 12 fps ja que alguns sistemes de videovigilància utilitzen aquestes característiques. És possible que en algun cas aparegui una latència entre la seqüència capturada i estabilitzada degut al filtrat dels paràmetres de moviment. Això, però, no impedeix que el processat es realitzi en temps real sinó que la seqüència de sortida estarà uns pocs segons enrederida respecte el vídeo capturat.

### 1.3 Diagrama de blocs de l'aplicació

Per tal d'assolir els objectius plantejats es dissenyarà una aplicació que respongui al diagrama de blocs de la figura 1.1. En primer lloc l'aplicació s'encarrega de realitzar l'estimació del moviment de la càmera entre dos frames consecutius. Aquest bloc l'implementarem de dues formes diferents per poder comparar quina ens dona millors resultats i amb el menor temps possible. La primera d'aquestes alternatives està basada amb una tècnica de descens de gradient multi nivell per tal d'evitar els mínims locals que es produeixen en aquestes situacions. A més a més, això també ens permetrà augmentar la velocitat de processament ja que primer es busca el mínim en una imatge amb menys resolució, és a dir, més petita. Posteriorment es va ampliant la imatge per refinar la posició d'aquest mínim. La segona alternativa està basada en la correlació bidimensional de les imatges. La correlació entre dos frames consecutius ens dona el desplaçament en x i en y d'un frame respecte l'altre.

Degut a la limitació del temps de càlcul per estabilitzar la imatge es fa impossible treballar amb tots els píxels de l'escena. Això fa que necessitem definir uns quants punts a partir dels quals es buscarà el moviment global entre dues imatges consecutives en una seqüència. Un punt singular és el que té un gradient elevat al seu voltant, és a dir, és aquell punt en que el nivell d'intensitat dels punts veïns varia considerablement. Aquests punts s'anomenen singulars perquè contenen una quantitat d'informació major que la resta, i per tant, són més adequats per treballar localment amb les imatges. Un dels mètodes més usats per tal de detectar punts singulars és del detector de Harris. Aquest detector requereix molts recursos i el temps de càlcul és molt elevat. És per això que altres autors han creat detectors més eficients i que consumeixen menys recursos però que mantenen un nivell de qualitat similar al detector de Harris.

Un cop haurem trobat els punts singulars haurem d'establir el moviment de cadascun d'ells entre dos fotogrames consecutius. Amb això aconseguirem estimar el moviment que ha sofert la càmera en l'espai de temps en que s'ha capturat un frame i el següent. Aquesta estimació ens servirà per determinar si el moviment de la càmera era l'esperat i per tant no hem de modificar la imatge, o si per contra, el moviment ha sofert alguna alteració i hem de compensar aquest moviment indesitjat. Per establir el moviment d'un punt entre dos frames consecutius existeixen diversos procediments entre els quals es poden destacar els algorismes basats en correlació.

Un cop estimat el moviment de la càmera s'haurà de diferenciar entre les vibracions i el moviment desitjat de la càmera per poder compensar les primeres. En el cas que el moviment de

la càmera sigui desconegut a priori, això es farà mitjançant un filtrat de l'estimació ja que es considerarà que el moviment desitjat de la càmera té unes components freqüencials molt menors que el moviment degut a les vibracions. Aquest filtrat ens introduirà una latència en la seqüència de sortida ja que per tal de poder filtrar necessitem el moviment d'una sèrie de frames consecutius.

Finalment es compensarà el moviment i es generarà una nova seqüència de vídeo lliure de vibracions. La compensació de moviment consisteix en reposicionar els píxels del frame a compensar per tal que el moviment final d'aquest frame respongui tan sols al model de moviment de la càmera sense la superposició de la vibració indesitjada.

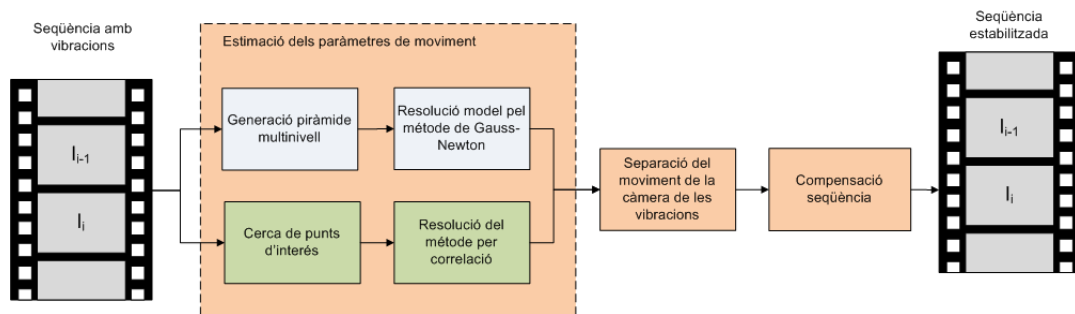


Figura 1.1: Esquema de blocs de l'aplicació

El següent document està estructurat de la següent forma. En el capítol 2 es dona una visió global de l'estat de l'art en la compensació de vibracions. També es fa una breu referència al treball que s'ha dut a terme en el TSC en relació en aquest tema. En els capítols 3 i 4 es tractaran les eines necessàries per resoldre l'estimació dels paràmetres de moviment. Aquests capítols corresponen als blocs del requadre '*Estimació dels paràmetres de moviment*' de la figura 1.1. En el capítol 5 s'estudiaran els blocs corresponents a la '*separació del moviment de la càmera de les vibracions*' i finalment la compensació de moviment necessària per eliminar les vibracions indesitjades de la seqüència de vídeo, '*Compensació seqüència*'. En el capítol 6 s'analitzaran els resultats obtinguts amb els software desenvolupat. Finalment en el capítol 7 es presentaran les conclusions del treball realitzat i es descriuran les pautes que es podrien seguir en un futur per continuar amb l'estudi de la compensació de vibracions en seqüències de videovigilància.

## Capítol 2

# Estat de l'art i treball previ

En aquest capítol analitzarem l'estat de l'art en la rama d'eliminació de moviment en seqüències de vídeo. Per començar, es realitzarà una breu pinzellada en la història de l'estabilització de seqüències de vídeo i posteriorment s'analitzaran diversos treballs sobre la compensació digital de seqüències de vídeo. Finalment exposarem breument el treball previ realitzat en el Grup de Processat d'Imatge del departament del TSC de la Universitat Politècnica de Catalunya.

### 2.1 Estat de l'art

Les tècniques de predicció i compensació de moviment porten uns quants anys sent investigades degut a la multitud d'aplicacions per a les que són necessàries. Per una banda són utilitzades en els estàndards de compressió de vídeo com MPEG-4. L'aplicació que ens interessa en aquest treball, però, consisteix en l'estabilització de seqüències de vídeo captades amb vibracions indesitjades.

A partir dels anys 50 es van començar a desenvolupar tècniques d'estabilització d'imatge. Aquests sistemes, anomenats sistemes d'estabilització d'imatge electrònics, utilitzaven sensors i actuadors per moure la lent de la càmera seguint els moviments indesitjats que capturaven els actuadors.

A finals de 1989 Mitsuaki Oshima et. al publicaren un treball [Osh89] on van desenvolupar el primer sistema d'estabilització d'imatge electrònic muntat sobre una videocàmera comercial. Les videocàmeres són dispositius molt exposats a les fluctuacions de vídeo i per això la demanda dels sistemes d'estabilització d'imatge ha incrementat amb l'ús d'aquestes. En aquest treball, en primer lloc van realitzar un estudi freqüencial del moviment aleatori de les càmeres de vídeo determinant que la major part de les vibracions es concentren a la freqüència d'1Hz. A aquesta freqüència aproximadament un 5% del moviment total en les càmeres era degut a les fluctuacions de les imatges. Després van realitzar un estudi amb diferents observadors dels vídeos enregistrats i van establir que els llindars de les fluctuacions perceptibles a 1Hz estan al voltant del 0,2% de fluctuació. Així van determinar que la taxa de supressió de vibracions era aproximadament de 15dB a 1Hz. El sistema muntat sobre la càmera consistia en una lent mòbil gràcies a dos actuadors i dos sensors per detectar la velocitat angular de la lent. Els actuadors que eliminaven la vibració de la lent eren controlats per un circuit en resposta al senyal detectat pels sensors. Un dels punts crítics d'aquesta aplicació estava en la operativitat de la càmera. L'eliminació del moviment de la lent dificultava el canvi de la direcció de captura de les imatges. És a dir, quan l'usuari canviava la direcció de filmació el sistema intentava corregir el moviment per continuar gravant en la mateixa direcció. Per solucionar aquest problema Oshima et al. van haver de desenvolupar un sistema per discriminar el moviment voluntari de la càmera del moviment

indesitjat. Aquest sistema utilitzava el fet que les fluctuacions involuntàries presentaven una constant de temps petita mentre que el moviment degut a l'usuari presentava una constant de temps molt major.

A partir d'aquest primer estudi altres autors han anat perfeccionant el sistema. Per exemple, Koichi Sato i co. van publicar [Sato93] on descriuen un sistema d'estabilització d'imatge electrònic mitjançant un prisma constituït per dos cristalls transparents amb un fluid comprés entre ells. Cadascun dels cristalls era rotat independentment per actuadors. D'aquesta forma aconseguien dirigir els rajos de llum incidents de forma que sense moure la lent podien eliminar les fluctuacions indesitjades. Actualment, gairebé totes les càmeres duen incorporats sistemes d'eliminació de vibracions. L'esmentat sistema ha estat adoptat per la marca Canon per produir objectius amb estabilitzador d'imatge. El sistema l'han anomenat "Vari-Angle Prism Image Stabilizer (VAP-IS)". En la web [Canon10] hi ha una breu descripció.

L'evolució de les tècniques d'estabilització electrònica s'ha apropiat poc a poc a l'estabilització de vídeo digital. En [Yong09] s'ha proposat un sistema d'estabilització on es calcula el vector de moviment de la càmera mitjançant giroscopis de gran precisió però per compensar les vibracions no utilitzen actuadors com en els casos anteriors, sinó que ho fan digitalment mitjançant processat d'imatge.

Els sistemes d'estabilització d'imatge electrònics requereixen càmeres de vídeo especialitzades i molt més cares que les normals. Una alternativa és l'estabilització d'imatge digital que consisteix en l'eliminació de les fluctuacions de la imatge digitalment mitjançant processat d'imatge a través de software. Així es poden utilitzar càmeres convencionals i per tant es redueix el cost final. A més a més, permet modificar el sistema d'estabilització d'imatge sense haver de modificar el hardware. Actualment les videocàmeres s'estan instal·lant en multitud d'aparells. Ordinadors personals, telèfons mòbils, ... Degut a això, l'interès per l'estabilització d'imatge en els últims anys ha fet que gran part de la investigació en processament d'imatge se centri en aquesta línia.

En [Shi07] es proposa un mètode basat en "Feature Block Searching" per eliminar el moviment i les vibracions de gran amplitud degudes a subjectes en moviment. A partir de blocs d'interès (cares, cotxes, senyals...), en primer lloc, estableixen el desplaçament en  $x$  i  $y$  i seguidament s'estabilitza el vídeo recentrant el bloc en la imatge. Per buscar aquestes translacions utilitzen el mètode de suma de diferències absolutes (SAD) per la seva simplicitat. En la mateixa línia en [Hu07] es calculen els vectors de moviment mitjançant el que anomenen "scale invariant features". Aquests punts són invariants respecte transformacions afins i canvis de lluminositat en les imatges. Seguidament utilitzen un filtrat gaussià i una aproximació parabòlica per detectar els moviments desitjats de la càmera i així poder compensar les vibracions. El mètode proposat presenta el principal avantatge que redueix les "zones negres" que es produeixen en les imatges degut a la compensació de moviment. Finalment proposen un mètode de mosaicing per reomplir aquestes "zones negres".

En el treball [Chao08] se segueix la mateixa línia d'estimació dels vectors de moviment basada en cerca de blocs característics (Feature-Block Motion Estimation). En aquest article els blocs són seleccionats del fons estàtic de l'escena mitjançant un detector de contorns. Només els blocs amb major quantitat de contorns són utilitzats per a l'estimació de moviment. A partir d'aquests blocs detecten les vibracions mitjançant la mitja de diferències quadrades (MSD). A més a més, en aquest treball també es proposa un mètode de restauració dels fotogrames després de la compensació. La reparació pot ser temporal o espacial en funció de la presència d'objectes en moviment o no en l'escena.

També en [Ondrej08] es realitza una estimació mitjançant el desplaçament d'un patró sobre una finestra de cerca per establir les translacions en la imatge. En aquest cas, igual que en l'anterior, el patró se selecciona fent un filtrat de la imatge previ per obtenir els contorns. L'aportació d'aquest treball consisteix en dividir la cerca en dues fases. En la primera s'estableix la translació de la imatge fent un delmat d'aquesta per tal que la velocitat de processat sigui substancialment

superior. En la segona fase es busca la transformació afí que estabilitza la seqüència a partir de la translació trobada en la primera fase.

Finalment trobem en [Yong10] un algoritme de Block Matching en tres passos. El mètode, però, només permet detectar translacions de la càmera. La proposta dels autors consisteix en una modificació de l'algoritme de cerca 2dlog que es pot observar en la figura 2.1 (a). L'algoritme original cerca la millor posició en 3 passos avaluant en cada cas 9 posicions candidates. En cada pas la regió de cerca es fa cada cop més petita. La novetat introduïda consisteix en avaluar 13 posicions enlloc de les 9 originals. Les posicions candidates addicionals se situen al voltant del centre formant un diamant com s'observa en la figura 2.1 (b). Suposant que la majoria de frames presenten un moviment reduït aquest mètode és capaç de detectar la posició en el primer pas sense haver d'executar els dos següents. Així l'eficiència de l'algoritme original és veu substancialment millorada.

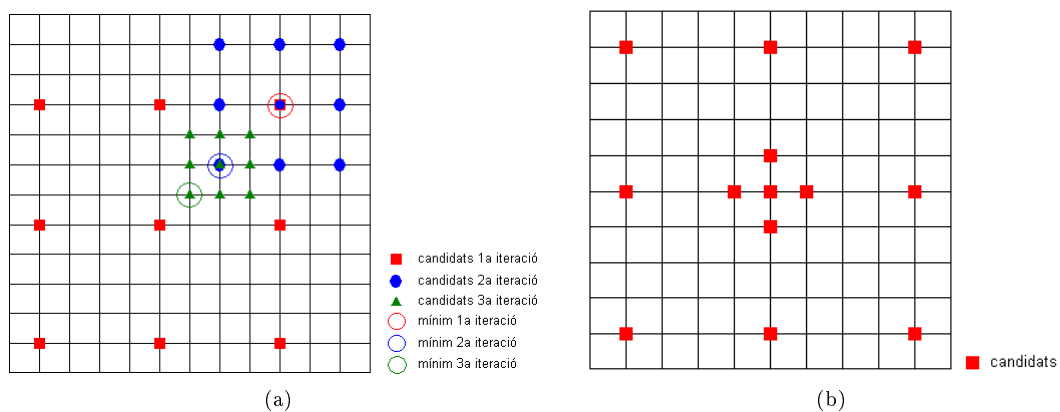


Figura 2.1: Comparació entre el mètode 2dlog original (a) i el nou mètode desenvolupat (b). En (b) si el mínim es troba en alguna de les 5 posicions del centre l'algoritme no realitza més passos

Un altre possible mètode d'estimació de moviment és el presentat en [Kwon07]. En aquest treball els seus autors proposen una estimació del moviment a partir de 8 vectors de moviment. En cada fotograma se seleccionen 8 subblocs que es divideixen en 2 grups tal com es mostra en la figura 2.2. A partir de la correlació dels vectors de moviment es distingeixen 4 possibles situacions:

- translació (mateixa direcció i magnitud)
- perspectiva (mateixa direcció però diferent magnitud)
- rotació respecte el centre (diferent direcció però mateixa magnitud)
- rotació (diferent direcció i magnitud)

En el primer cas la compensació es realitza amb el model translacional, en els restants compensen tant la rotació com la perspectiva mitjançant la homografia entre els dos frames que s'estan comparant a partir de les correspondències dels 8 subblocs seleccionats.



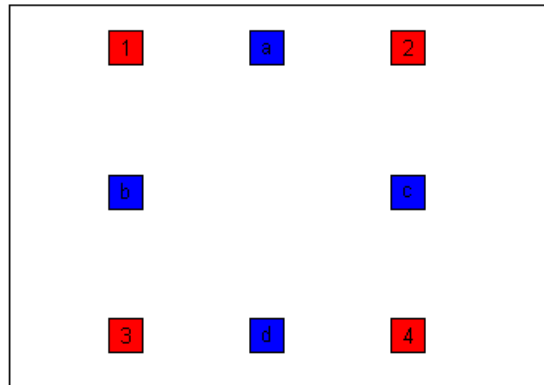


Figura 2.2: Agrupació dels 8 blocs per a l'estimació de moviment.

Seguint la línia del treball anterior, en [Bat08] es realitza una estimació de moviment mitjançant un filtrat dels vectors de moviment. En primer lloc s'estimen els vectors de moviment locals dividint la imatge en diferents blocs. L'aportació dels autors consisteix en la proposta de nous mètodes de pre-filtrat dels vectors de moviment que permeten descartar els que no segueixen el moviment general de la imatge. En segon lloc realitzen el que anomenen “Memory Filter” que consisteix en descartar els vectors que no segueixen el moviment dels frames precedents. Finalment realitzen una estimació mitjançant un mètode de mínims quadrats amb el que són capaços d'oferir robustesa enfront del soroll present en la seqüència. El mètode també resulta efectiu davant la presència d'objectes mòbils en l'escena.

En [Yuan07] es proposa un altre mètode d'estimació de moviment en dos passos. En el primer pas es realitza una transformació log-polar multi-resolució. Seguidament utilitzen un mètode de descens de gradient per refinar l'estimació. El principal avantatge de la transformació log-polar resideix en el fet que les rotacions i escalats en la imatge original es converteixen en translacions fàcilment detectables per correlació en la imatge transformada.

Altres autors han proposat l'utilització d'algoritmes genètics per a l'estimació de moviment com és el cas de [Nait07]. Aquests mètodes presenten l'avantatge que detecten i compensen el moviment correctament fins i tot en situacions amb una SNR molt baixa. El principal inconvenient és que al tractar-se d'un algoritme iteratiu la velocitat de convergència és molt baixa. A més a més, l'algoritme presentat en aquest treball només compensa vibracions translacionals.

Un altre possible mètode d'estimació de moviment és el presentat en [Bosco08]. A diferència de tots els altres treballs, en aquest no es realitza cap correlació per estimar l'estimació dels vectors de moviment sinó que s'utilitzen les corbes característiques o signatures dels fotogrames. Cada fotograma genera dues signatures  $C_x$  i  $C_y$  que són la suma de la intensitat dels píxels normalitzada en l'eix  $x$  i  $y$  respectivament. És a dir, la signatura es pot definir com una integral respecte cadascun dels eixos de la imatge. Per tal d'estimar el moviment busquen el desplaçament entre signatures de fotogrames consecutius. El principal avantatge que confereix aquest mètode és la robustesa enfront a canvis bruscos d'il·luminació. La signatura, encara que presenta diferent amplitud, té la mateixa forma fins i tot quan hi ha un canvi d'il·luminació sobtat en la seqüència.

## 2.2 Treball previ

En el grup d'imatge del departament del TSC s'ha desenvolupat un software que realitza l'estimació de moviment entre dos frames d'una seqüència de vídeo. El software utilitza l'algoritme de Gauss-Newton per determinar el vector de moviment entre dos fotogrames de la seqüència.

A més també permet seleccionar el tipus de transformació a aplicar a la imatge. Les transformacions disponibles són el model translacional (2 variables), la transformació afí (6 paràmetres) o la transformació perspectiva (10 paràmetres).

El software realitza una estimació del vector de moviment i calcula els paràmetres de la transformació escollida. La transformació a utilitzar s'haurà d'escollir en funció de la naturalesa del moviment de la seqüència d'imatges. Evidentment, com més paràmetres tingui el model escollit més augmentarà el cost computacional i per tant el temps de processat. A més a més, l'error d'estimació també serà superior com més paràmetres s'hagin d'estimar. Tenint tot això en compte, sempre haurem d'escollir el model que amb els mínims paràmetres descrigui de forma acceptable el moviment de la càmera.

L'algoritme de Gauss-Newton (GNA) dona una solució numèrica per a la minimització d'una funció, normalment no linear, sobre un espai de paràmetres d'aquesta. El GNA es un mètode molt paregut a l'algoritme de descens de gradient. En el cas que ens ocupa els paràmetres a optimitzar són els corresponents al model de transformació escollit. Si, per exemple, se selecciona el model de transformació afí l'algoritme haurà d'establir els 6 paràmetres que regeixen aquesta transformació,

$$x_1 = a_1 \times x_0 + a_2 \times y_0 + a_3$$

$$y_1 = a_4 \times x_0 + a_5 \times y_0 + a_6$$

Els paràmetres que haurà de trobar el mètode són  $\bar{A} = [a_1 a_2 a_3 a_4 a_5 a_6]$ . La funció que volem minimitzar és l'histograma de l'error entre la imatge original i la imatge transformada. Així doncs, quan aquest histograma sigui mínim, els paràmetres  $\bar{A}$  seran la solució òptima.

L'algoritme de Gauss-Newton és un mètode iteratiu. Com qualsevol mètode iteratiu requereix una inicialització dels paràmetres per començar a buscar la solució de la minimització. Com més propers siguin els paràmetres inicials a la solució òptima menys trigarà l'algoritme a trobar-la. A més a més, com qualsevol altre, aquest mètode podria arribar a donar una solució errònia en cas que caigui en un mínim local. Per solucionar aquests dos problemes l'aplicació desenvolupada presenta dues solucions complementàries.

Per una banda s'utilitzen els paràmetres de la iteració anterior per inicialitzar l'algoritme. Degut a la naturalesa de l'aplicació, la solució òptima dels paràmetres del model de transformació seran molt pareguts entre dos parells de frames consecutius. Així, un cop determinats els paràmetres entre els frames  $i - 2$  i  $i - 1$ , aquesta serà una molt bona inicialització per començar a buscar els paràmetres entre els frames  $i - 1$  i  $i$ . Aquesta inicialització incrementarà la velocitat de convergència de l'algoritme, ja que la solució que minimitza la funció serà molt propera als paràmetres inicials.

Per altra banda, i per solucionar el problema dels mínims locals, s'ha optat per un mètode multi nivell en 3 passos. Per calcular els paràmetres de la transformació d'una imatge de  $N \times M$  píxels en primer lloc es redueix la imatge a un tamany  $\frac{N}{4} \times \frac{M}{4}$ . Amb aquesta nova imatge es calcula la solució que minimitza l'histograma de l'error. Per trobar la solució en aquest primer nivell s'utilitza la solució obtinguda en la iteració anterior tal com acabem de comentar. Un cop trobada la solució òptima per  $\bar{A}$  en aquest nivell es comença a buscar la solució amb les imatges del següent nivell que tenen un tamany de  $\frac{N}{2} \times \frac{M}{2}$  píxels. Per inicialitzar aquest nivell, òbviament, s'utilitza la solució obtinguda en el nivell precedent multiplicant per 2 alguns paràmetres com podrien ser els que descriuen les translacions. Finalment, amb la solució d'aquest nivell s'inicialitza el tercer i últim amb les imatges originals de  $N \times M$  píxels per trobar la solució final per als paràmetres de la transformació seleccionada. En la figura 2.3 hi ha una representació gràfica de la piràmide de 3 imatges utilitzades per calcular la solució amb el GNA. En la secció 4.3.1 del capítol 4 hi ha una descripció més detallada d'aquest mètode.

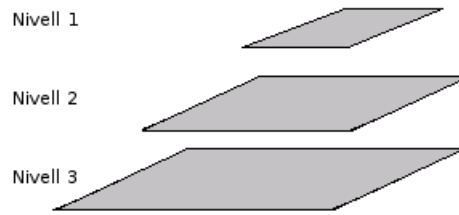


Figura 2.3: Piràmide d'imatges. Primer es busca el mínim global en la imatge de menor resolució i posteriorment es refina amb les imatges dels nivells inferiors.

El mètode d'estimació dels paràmetres de moviment fa que el software sigui molt poc eficient i per tant, no és possible utilitzar-lo per eliminar vibracions en temps real (entre 12 i 25 fps) ja que la velocitat d'execució és de 0.5 fps aproximadament. Malgrat que es basa en el descens de gradient, al tractar-se d'un algorisme iteratiu multi nivell el cost computacional és molt elevat, i fins i tot, amb els processadors més moderns no n'hi ha prou per realitzar la compensació a temps real. Això representa un impediment a l'hora d'implementar tècniques de compensació de moviment software en sistemes comercials.

## Capítol 3

# Detecció de punts d'interès

Un cop presentat el treball a realitzar continuarem amb la descripció dels diferents blocs que intervenen en l'estimació de moviment. Com ja s'ha comentat en la introducció, per tal de realitzar l'estimació de moviment existeixen dues alternatives. La primera consisteix en un mètode de descens de gradient que s'explicarà en el capítol 4. La segona consisteix en realitzar una estimació mitjançant la correlació entre dos frames consecutius. En aquest segon mètode, un dels punts crítics de l'aplicació és la selecció de punts representatius de la imatge. En la figura 3.1 podem observar un diagrama de blocs de l'aplicació amb la ubicació de la detecció de punts d'interès marcada en vermell.

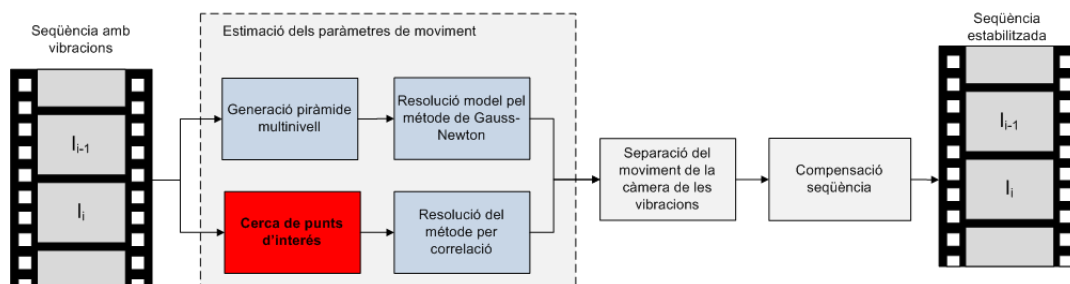


Figura 3.1: Diagrama de blocs de l'estimació dels paràmetres de moviment. En vermell està representat el bloc del que tracta aquest capítol.

Degut a la limitació del temps de processat per poder aconseguir velocitats acceptables es fa impossible treballar amb tots els píxels de la imatge. Molts dels mètodes de compensació de moviment utilitzen només un conjunt de píxels representatius de la imatge per realitzar aquest processat. Per tant, és necessari un mètode de selecció d'aquests punts més representatius. En aquest capítol s'estudiaran diverses alternatives per a la cerca de punts d'interès i es proposarà un detector adequat per a l'aplicació que ens ocupa.

En processat d'imatge és bastant comú utilitzar el que s'anomena 'features' o 'borders' per tal d'establir correspondències entre imatges. Els punts d'interès són punts de la imatge on es produeix un canvi d'intensitat brusc de forma que és més fàcil crear mesures de semblança per a aquests punts. Exemples de punts d'interès en una imatge són els contorns d'una figura o, encara millor els vèrtexs. En aquest capítol es presenten dos detectors de punts d'interès ben documentats els quals ens serviran per desenvolupar un detector que s'adapti a les necessitats de l'aplicació. Posteriorment s'explicarà el detector desenvolupat per a la nostra aplicació.

### 3.1 El detector de Harris

Un dels detectors de punts d'interès més utilitzats és el detector de Harris [Harris98]. Aquest detector és invariant a la rotació, escala, canvis d'il·luminació i soroll en la imatge. Aquests trets són els que el fan interessant a l'hora de buscar punts en les imatges. El fet que el detector sigui invariant significa que encara que fem canvis en la imatge els punts trobats són els mateixos. És a dir, si tenim un quadrat en la imatge i el detector troba un punt d'interès en un dels vèrtexs, encara que rotem, escalem o modifiquem l'il·luminació de la imatge el detector ens trobarà el mateix punt. En la figura 3.2 podem veure un exemple de la invariància del detector en diferents formes. Per provar aquesta invariància s'ha creat una imatge amb diferents figures i s'han buscat els punts d'interès mitjançant una implementació en Matlab del detector de Harris. Els resultats obtinguts han estat els esperats, és a dir, els punts continuen sent els mateixos encara que fem els canvis descrits anteriorment en la imatge original.

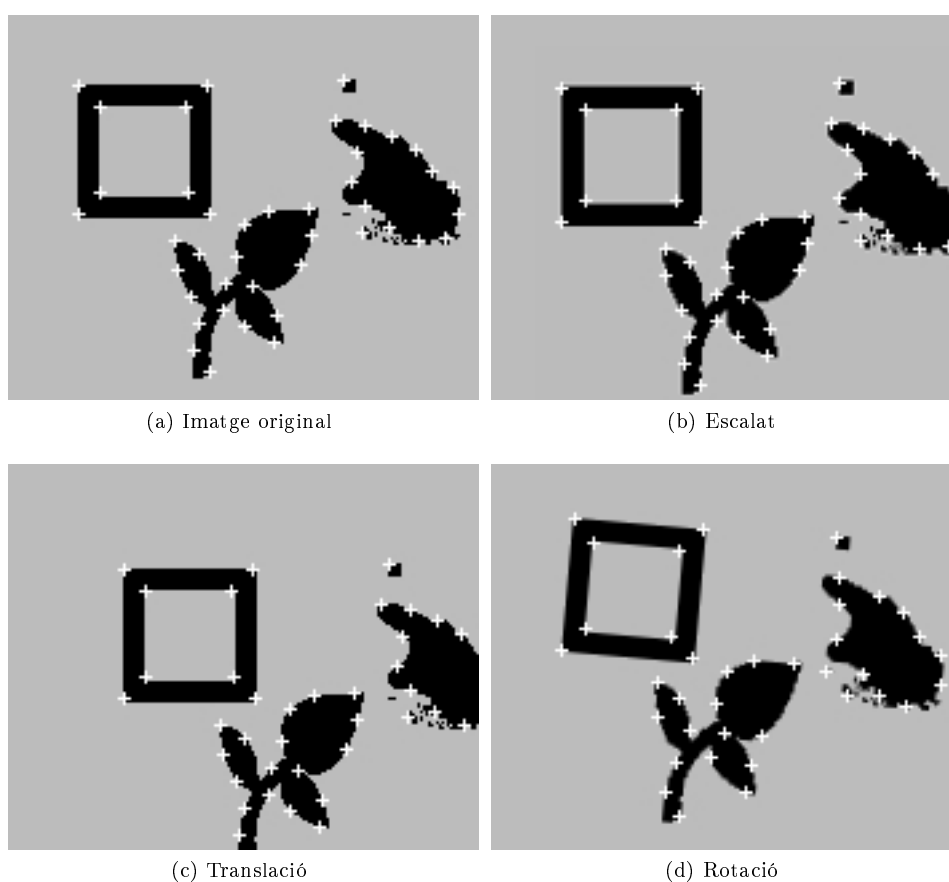


Figura 3.2: Transformacions en la imatge per comprovar la invariància del detector de Harris. En (a) tenim la imatge original. En (b) s'ha fet un escalat de la imatge. Observant els punts detectats es pot veure la invariància del detector respecte un canvi d'escala. En (c) s'ha fet una translació de la imatge i en (d) s'ha aplicat una rotació. Els punts trobats continuen sent pràcticament els mateixos en tots els casos.

El detector de Harris es basa en l'autocorrelació local entre una imatge i diverses subimatges desplaçades en múltiples direccions. Donat un desplaçament  $(\Delta x, \Delta y)$  i un punt  $(x, y)$ , la funció d'autocorrelació es defineix com,

$$c(x, y) = \sum_W [I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)]^2 \quad (3.1)$$

on  $W$  representa tots els punts dintre d'una finestra gaussiana. La imatge desplaçada es pot aproximar pels termes del polinomi de Taylor fins a ordre 1,

$$I(x_i + \Delta x, y_i + \Delta y) \approx I(x_i, y_i) + [I_x(x_i, y_i) \ I_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (3.2)$$

on  $I_x$  i  $I_y$  són les derivades parcials en  $x$  i  $y$  respectivament. Substituint l'aproximació (3.2) en (3.1) obtenim,

$$\begin{aligned} c(x, y) &= \sum_W \left( I(x_i, y_i) - I(x_i, y_i) - [I_x(x_i, y_i) \ I_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \\ &= \sum_W \left( [I_x(x_i, y_i) \ I_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \\ &= [\Delta x \ \Delta y] \begin{bmatrix} \sum_W (I_x(x_i, y_i))^2 & \sum_W I_x(x_i, y_i) I_y(x_i, y_i) \\ \sum_W I_x(x_i, y_i) I_y(x_i, y_i) & \sum_W (I_y(x_i, y_i))^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\ &= [\Delta x \ \Delta y] C(x, y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \end{aligned} \quad (3.3)$$

La matriu  $C(x, y)$  conté informació sobre l'estructura d'intensitat al voltant del punt  $(x, y)$  per a desplaçaments  $(\Delta x, \Delta y)$  petits. Un punt d'interès està caracteritzat per variacions grans d'intensitat en totes les direccions al voltant del punt. Analitzant els valors propis de la matriu es pot arribar a la conclusió que:

1. Si  $\lambda_1 \approx 0$  i  $\lambda_2 \approx 0$  el punt  $(x, y)$  no es cap punt d'interès.
2. Si  $\lambda_1 \approx 0$  i  $\lambda_2$  un valor elevat, en el punt  $(x, y)$  hi ha un borde.
3. Si tant  $\lambda_1$  com  $\lambda_2$  presenten un valor elevat, el punt  $(x, y)$  és un vèrtex i per tant un punt d'interès.

Harris i Stephens van descobrir que no és necessari calcular els valors propis de la matriu i van suggerir la funció  $M_c$ ,

$$M_c = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 = \det(A) - k \text{trace}^2(A)$$

on  $k$  és un paràmetre que s'ha determinat empíricament i segons la literatura el valor ha de ser entre 0.04 i 0.15 per al correcte funcionament.

La figura 3.3 a la pàgina següent mostra els punts trobats en una imatge mitjançant el detector de Harris. La rutina s'ha implementat en Matlab per poder comprovar el seu funcionament. Es pot observar en ella que tots els punts detectats presenten una gran variació del nivell d'intensitat. Paral·lelament podríem definir aquest punts com els que tenen major quantia d'informació. Donat un punt corresponent a una textura és molt més probable que els píxels



Figura 3.3: Punts d'interès trobats mitjançant el detector de Harris.

veïns tinguin més o menys la mateixa intensitat. Quan la intensitat canvia bruscament és degut a un canvi de textura. És molt més fàcil trobar la correlació d'aquests punts ja que la funció de correlació presentarà un màxim molt més diferenciat dels de la resta.

En la figura 3.4 podem observar la importància de la cerca dels punts d'interès. Primerament s'ha extret un template de  $50 \times 50$  de la imatge de la figura 3.3. Aquest template s'ha escollit de forma que no contingui cap punt d'interès. La regió de cerca s'ha extret de la mateixa imatge i d'un tamany de  $100 \times 100$  píxels. En la figura 3.4 (b) podem observar la correlació entre el template i la regió de cerca. Seguidament s'ha extret un altre template de  $50 \times 50$  i una regió de cerca que conté el template (c). Aquest segon template conté algun dels punts d'interès trobats pel detector de Harris. En (d) podem observar la correlació entre el template i la regió de cerca. Per tal d'establir la posició del template dintre de la regió de cerca cal buscar el màxim de la correlació. En (d) podem distingir clarament el màxim de la correlació ja que aquest presenta un pic en la posició en que el template casa correctament dintre de la regió de cerca. En la figura (b), en canvi, no podem distingir el màxim ja que hi ha molts punts amb una correlació molt pareguda. Així doncs, si en el template tenim un punt d'interès la precisió de la localització serà molt més acurada.

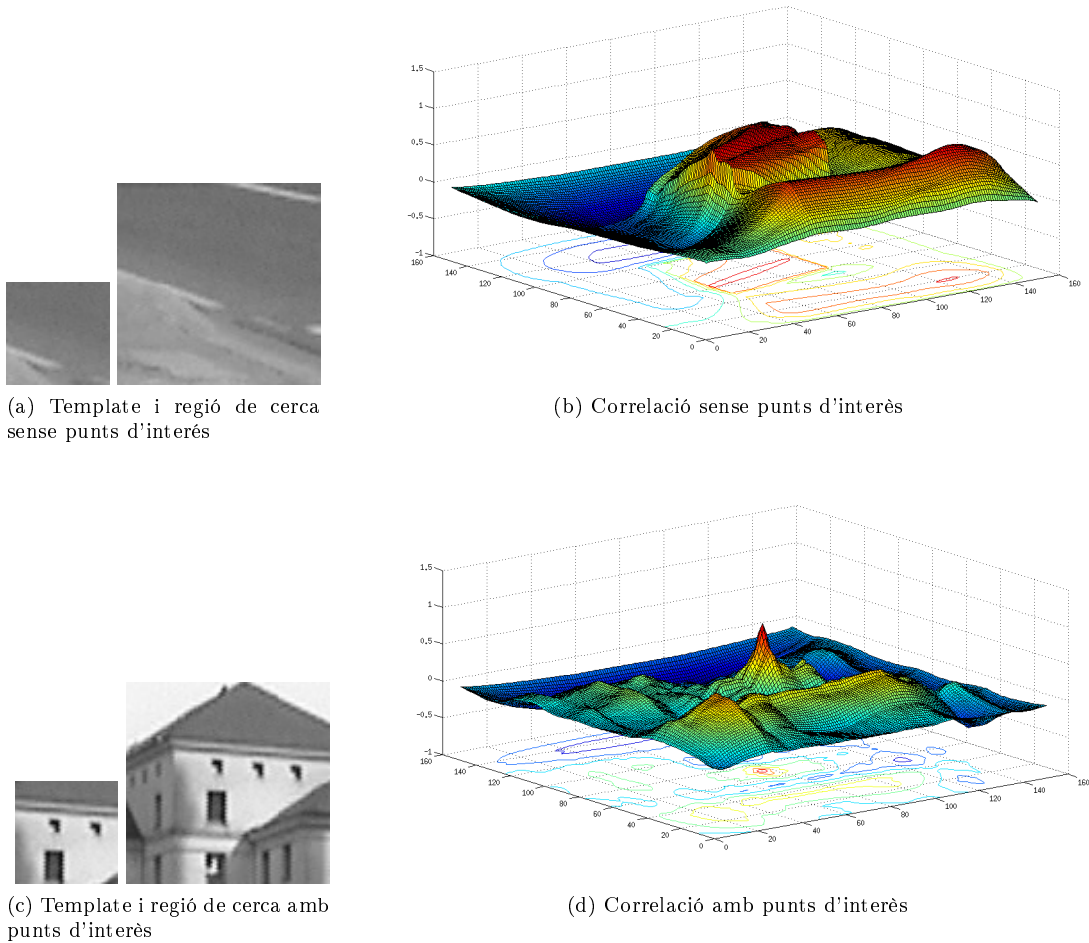


Figura 3.4: Comparació de la correlació. En (a) tenim una porció de la imatge de la figura o template sense cap punt d'interès i la regió de cerca. En (b) tenim la correlació normalitzada entre el template i la regió de cerca de (a). En (c) tenim el template i la regió de cerca amb punts d'interès i en (d) tenim la correlació normalitzada entre el template i la regió de cerca de (c).

### 3.2 El detector de Trajkovic-Headley

El detector de Harris és poc eficient ja que la càrrega computacional és molt elevada. Per tal de buscar els punts d'interès s'ha de calcular, per a cada punt de la imatge, l'aproximació de la matriu de covariància descrita en 3.3 i calcular el determinant i la traça de la matriu al quadrat. Un dels nostres objectius consisteix en l'augment de la velocitat de processament de les seqüències de vídeo. És per això que haurem de buscar detectors de punts amb menys càrrega computacional. Altres autors [Shi94, Smith97] han investigat en aquesta línia per obtenir detectors de punts d'interès més eficients. En l'aplicació que estem desenvolupant tindrem en compte el detector desenvolupat per Trajkovic i Headley [Tra98] ja que és tracta d'uns dels més eficients en la cerca de punts d'interès.



Aquest detector comprova els canvis d'intensitat en un punt observant els píxels veïns. Si prenem  $c = (x_c, y_c)$  com el píxel que estem analitzant,  $p = (x_p, y_p) \in P$  els punts pertanyents a la circumferència  $P$  centrada en  $c$ , i  $p' = (x_{p'}, y_{p'}) \in P$  el punt diametralment oposat a  $p$ , definim la següent funció (Corner Response Function - CRF),

$$r(c) = \min_{p \in P} [(I(p) - I(c))^2 + (I(p') - I(c))^2] \quad (3.4)$$

En la figura 3.5 trobem una representació dels punts escollits en funció del diàmetre de la circumferència  $P$ . Els píxels que pertanyen a la circumferència s'han marcat en gris. Per calcular  $r(c)$  es van agafant aquests píxels de dos en dos, tal com mostren les aspes vermelles, i s'aplica l'equació (3.4).

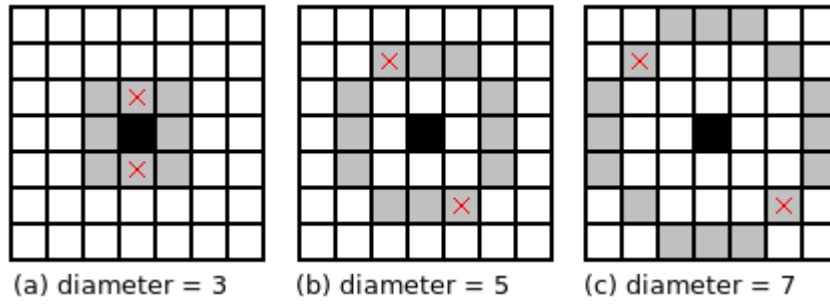


Figura 3.5: El píxel negre és  $c$ . Els píxels grisos són els punts que pertanyen a la circumferència  $P$ . Els píxels marcats amb una aspa vermella representen possibles combinacions de  $p$  i  $p'$ .

La funció  $r(c)$  prendrà valors elevats quan els punts veïns de  $c$  siguin molt diferents. Per tal de determinar que el punt en qüestió és un punt d'interès haurem de comparar el valor obtingut amb un llindar. Per a cada seqüència s'haurà d'analitzar quin és el millor nivell llindar. Si  $\lambda_d$  és el llindar de detecció, es defineix  $I$  com el conjunt de punts d'interès que compleixen l'equació,

$$r(c) > \lambda_d \iff c \in I \quad (3.5)$$

En la figura 3.6 podem observar els punts trobats sobre un frame de la seqüència de prova. S'han realitzat dues realitzacions de la cerca per a llindars diferents. A més del llindar, el detector també requereix un nombre màxim de punts a trobar per tal de no fer la cerca excessivament llarga. Si elegim el llindar de detecció massa baix, s'assoleix el nombre màxim de punts al principi i selecciona gairebé tots els punts que analitza (a). Si per contra, elevem el valor del llindar per evitar aquest efecte, llavors només troba punts situats en les zones on hi ha molta variació d'intensitat (b). En aquest cas la silueta de l'arbre.

Un altre dels paràmetres que podem variar en el detector és el diàmetre de la circumferència  $P$ . En la taula 3.1 hi ha el nombre de punts trobats pel detector en funció del diàmetre. Es pot observar en aquesta taula la dependència que existeix entre el diàmetre de cerca i el llindar. Lògicament com més gran sigui el diàmetre més gran serà el valor de la funció  $r(c)$  descrita en 3.4 ja que els píxels que comparem estan més lluny del centre. Amb un mateix valor de llindar trobarem més punts com major sigui el diàmetre de la circumferència  $P$ .



(a) Llindar de detecció = 500

(b) Llindar de detecció = 2000

Figura 3.6: Detector original. Si el llindar és massa petit, els punts trobats estan tots a la part superior perquè es comença a analitzar la imatge de dalt a baix i d'esquerra a dreta (a). Si elevem el llindar només troba els punts amb variació d'intensitat molt elevada (b).

Diàmetre	Llindar	Punts trobats
3	500	380
3	1000	115
3	2000	12
5	500	>1000
5	1000	561
5	2000	113
7	500	>1000
7	1000	>1000
7	2000	859

Taula 3.1: Nombre de punts trobats en funció dels paràmetres del detector

En la figura 3.7 es poden observar els punts detectats en funció del diàmetre de la circumferència per calcular la funció CRF. Com hem vist en la taula 3.1 al variar el diàmetre també s'ha de variar el llindar de detecció si es vol obtenir un nombre de punts més o menys constant. Si ens fixem en aquestes figures s'observa que malgrat poder variar els paràmetres del detector per tenir els punts ben distribuïts és molt difícil aconseguir trobar punts al llarg de tota la imatge. Si el llindar és massa petit gairebé es donen per bons tots els punts avaluats mentre que si el llindar és massa gran només es troben els punts amb una variació molt forta i això fa que aquests no quedin ben distribuïts per tota la imatge. A més a més, en les seqüències de vídeo que tractarem no es pot assegurar que la textura serà més o menys constant durant tota la seqüència ja que la càmera pot estar en moviment. Això faria que el llindar de detecció s'hagués d'anar ajustant en cada iteració per obtenir el valor òptim.

(a) Diàmetre  $P = 3$ , llindar de detecció 500(b) Diàmetre  $P = 5$ , llindar de detecció = 1000(c) Diàmetre  $P = 7$ , llindar de detecció = 2000

Figura 3.7: Comparació del diàmetre de la circumferència  $P$  per al càlcul de la funció  $r(\vec{c})$ . A més del diàmetre també s'ha variat el llindar de detecció per aconseguir tenir els punts més o menys ben distribuïts per tota la imatge

### 3.3 Detector desenvolupat

En l'aplicació que ens ocupa necessitem que els punts trobats estiguin distribuïts per tota la imatge. A més, tal com s'ha comentat, necessitem un detector més eficient que el detector de Harris. També cal remarcar que és convenient que el nombre de punts i la seva distribució no varii excessivament en funció dels paràmetres del detector. Per tal de calcular els paràmetres de moviment només es tenen en compte petites regions amb el centre en els punts d'interès trobats. Per tal de representar el moviment global de la imatge és necessari que aquestes regions es trobin ben repartides per tota la imatge. Com hem pogut observar en la figura 3.6 amb el detector de Trajkovic-Headley original és molt difícil que els punts es trobin distribuïts per tota l'escena. Per tal de millorar el funcionament d'aquest detector per a la nostra aplicació introduïrem dues modificacions a l'algoritme. Les modificacions no són complementaries i per tant ens haurem de quedar amb la que ens doni millors resultats.

Per tal de descriure el moviment d'una imatge la podem dividir en petites regions cadascuna d'elles amb el mateix moviment. El moviment global de la imatge es calcularà a partir del promitjat del moviment d'aquestes regions. Els dos mètodes proposats a continuació consisteixen en calcular el moviment d'aquestes regions.

### Establiment de distància mínima

La primera modificació consistirà en considerar una distància mínima entre els punts d'interès. És a dir, entre dos punts trobats haurà d'existir una distància mínima predefinida. Tot i que aquesta modificació no divideix la imatge en petites regions, amb l'establiment de la distància mínima el resultat que s'aconsegueix es buscar només un punt d'interès dintre de cada subregió.

Aquesta modificació ens aportarà per una banda una millora en l'eficiència de l'algoritme, ja que quan fixem un punt no calculem  $r(\vec{c})$  per als seus veïns que es troben a una distància menor que la mínima predefinida. Malgrat que no podem assegurar que els punts es trobaran distribuïts per tota la imatge, ajustant els paràmetres del detector (llindar i diàmetre de la circumferència  $P$ ) podem arribar a tenir els punts ben distribuïts.

En la taula 3.2 presentem el nombre de punts trobats variant els paràmetres esmenats. Seguidament, en la figura 3.8 observem tres realitzacions del detector amb diferents valors per a la distància mínima.

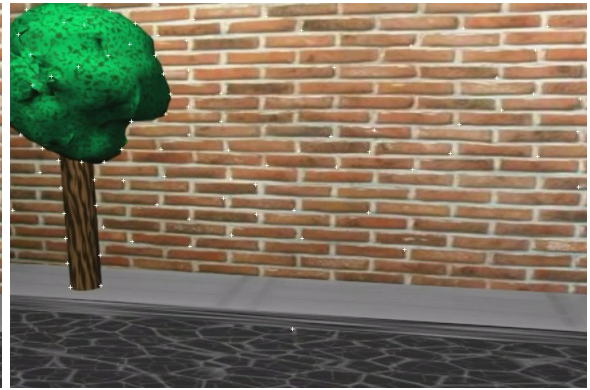
Diàmetre	Distància	Llindar	Punts trobats
3	1	2000	12
3	25	500	56
3	25	1000	19
3	50	500	30
5	1	2000	113
5	25	500	178
5	25	1000	85
5	50	500	85
7	1	2000	859
7	25	1000	238
7	25	2000	175
7	50	1000	105

Taula 3.2: Nombre de punts trobats en funció dels paràmetres del detector

Els valors del detector s'han escollit al voltant d'aquests valors ja que el nombre de punts necessaris per realitzar el posterior processat es troba al voltant de 100. En la figura 3.8 podem observar algunes de les realitzacions que trobem a la taula anterior.



(a) Distància mínima = 1



(b) Distància mínima = 25



(c) Distància mínima = 50

Figura 3.8: Primera modificació del detector. Establiment de distància mínima

Per tal d'elegir els paràmetres que millor s'adapten a la nostra aplicació també hem de tenir en compte que com major sigui el llindar més píxels veïns s'hauran de contemplar. Amb aquesta consideració i tenint en compte les imatges observades, els valors que escollirem per realitzar les proves són:  $d = 5$ ;  $\lambda_d = 1000$ ;  $d_{min} = 25$ .

### Subdivisió de la imatge

En segon lloc modificarem el detector de Trajkovic-Headley dividint la imatge prèviament en tantes regions com punts volem trobar. En cada regió buscarem el punt que presenti el màxim valor de  $r(\vec{c})$ . Amb aquesta modificació assegurem que els punts queden distribuïts per tota la imatge. A més a més, només busquem els punts necessaris per als següents blocs funcionals de l'aplicació.

Tant en el detector original com amb la primera modificació no podem controlar el nombre de punts que troba el detector, sinó que només podem establir un nombre màxim de punts. Amb aquesta modificació assegurem, per una banda, que només busquem els punts necessaris i per tant no n'hem de descartar cap després d'haver-lo trobat. Per altra banda els punts queden igualment distribuïts per tota la imatge, i a més seleccionem el punt que millor representa la regió en qüestió.

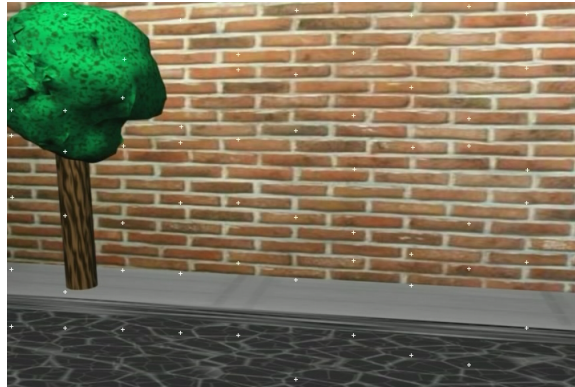


Figura 3.9: Segona modificació. S'ha dividit la imatge en  $n \times m$  regions i s'ha buscat el punts amb màxima variació en cada regió.

En la figura 3.9 s'observa una realització d'aquesta modificació i es pot observar com, efectivament, la imatge s'ha dividit en diferents regions, i per cada regió tenim el punt amb major variació d'intensitat. Una altre avantatge d'aquesta modificació l'analitzarem en el punt 4.3 a la pàgina 33, on calcularem els paràmetres de moviment a partir de triades de punts trobades pel detector.

En el capítol 6 analitzarem el comportament d'aquestes dues modificacions i s'avaluarà quina d'elles dona millors resultats per a l'aplicació que estem desenvolupant.

## Capítol 4

# Obtenció dels paràmetres de moviment

Continuant amb el càlcul dels paràmetres de moviment en aquest capítol s'explicaran les dues formes d'estimació que s'han desenvolupat. Per tenir una idea del punt en que ens trobem en la figura 4.1 podem observar els blocs corresponents a l'estimació de moviment dintre del diagrama de blocs general de l'aplicació.

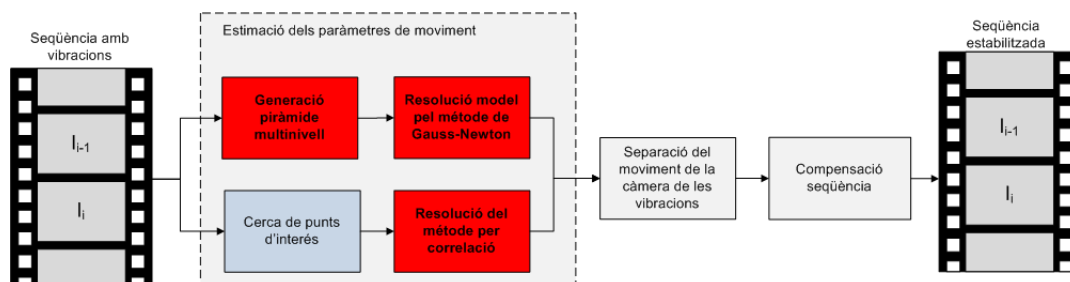


Figura 4.1: Diagrama de blocs de l'aplicació. En vermell hi ha els blocs dels que tracta aquest capítol.

El model de transformació expressa la deformació que existeix entre dues imatges d'una seqüència quan són captades pel sensor d'una càmera. Els models més utilitzats són el translacional, que només contempla desplaçaments; el model afí, que a més de transformacions també té en compte canvis d'escala i deformacions en algun dels seus eixos; i la transformació perspectiva, que com el seu nom indica, contempla tot tipus de deformacions de perspectiva.

El tret característic més important d'aquests models és el nombre de paràmetres utilitzats per poder-los representar. En processat d'imatge, per tal d'establir la relació entre dues imatges utilitzant un model determinat, és necessari calcular tots els paràmetres corresponents a la transformació utilitzada. Per tal de decidir quin model utilitzarem haurem de tenir en compte que aquest haurà de poder representar totes les transformacions sofertes per la imatge. A més haurem d'escollir el model que, complint la condició anterior, tingui el nombre mínim de paràmetres per tal de reduir la complexitat dels càlculs i augmentar l'eficiència de l'estimació.

El principi bàsic de l'estimació de moviment consisteix en establir la relació entre dos frames d'una seqüència de vídeo:

$$I_n(x, y) = \tilde{I}_{n-1}(f(x, y), g(x, y)) = \tilde{I}_{n-1}(x', y') \quad (4.1)$$

En l'equació (4.1)  $x' = f(x, y)$  i  $y' = g(x, y)$  dependran del model utilitzat. En el següent punt descriurem els diferents models de moviment.

## 4.1 Models de transformació

Una transformació en processat d'imatge consisteix en mapejar el píxel  $(x_0, y_0)$  corresponent a una imatge en la posició  $(x_1, y_1)$  corresponent a la imatge transformada. En la figura 4.2 es pot observar aquest efecte.

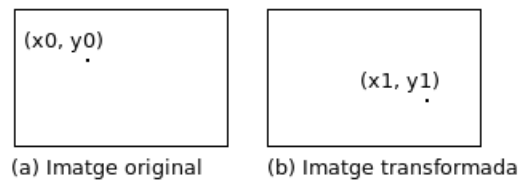


Figura 4.2: Transformació d'una imatge

En aquest punt s'analitzaran diferents models de transformació i s'escollirà el més adient per a l'aplicació que ens ocupa. La figura 4.3 mostra la imatge de mostra que prendrem per observar les deformacions sofertes per cadascuna de les transformacions que estudiarem.

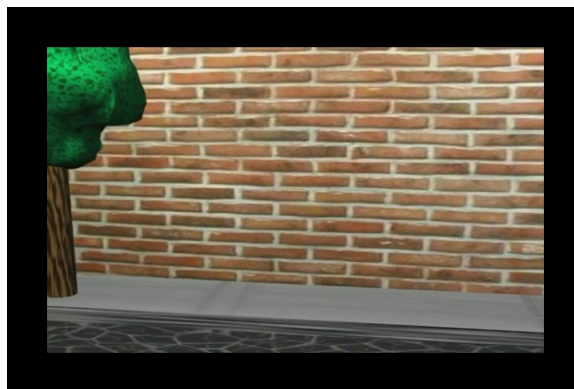


Figura 4.3: Imatge original per observar l'efecte de les transformacions

A partir d'aquesta imatge aplicarem cadascuna de les transformacions per observar de forma gràfica els efectes que produeixen. A més aquestes imatges ens ajudaran a comprendre en un simple cop de vista l'efecte de la transformació.

## 4.2 Transformació translacional

La transformació translacional és la més simple de totes. Com el seu nom indica només cerca una translació en els píxels de la imatge. La translació pot ser en sentit horitzontal o vertical. Aquest model de moviment es descriu mitjançant les següents equacions:



$$f(x, y) = x - \hat{u}_i$$

$$g(x, y) = y - \hat{v}_i$$

on  $\hat{u}_i$  és l'estimació de la translació horitzontal i  $\hat{v}_i$  és l'estimació de la translació vertical. El vector format per l'estimació  $(\hat{u}_i, \hat{v}_i)$  s'anomena vector de moviment i es calcula mitjançant l'estimació de moviment entre el frame  $I_n$  i  $\tilde{I}_{n-1}$ . Aquest model de transformació només té dos paràmetres i és el que requereix menys cost computacional. Per altra banda, només pot representar translacions de la imatge. En la figura 4.4 es pot observar l'efecte produït per aquesta transformació.

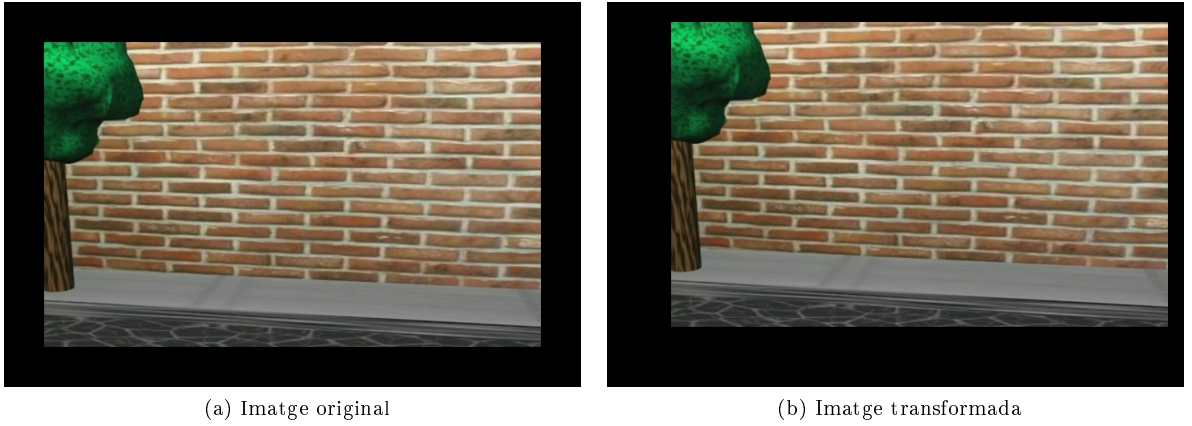


Figura 4.4: Transformació translacional. La transformació aplicada ha estat  $(30, -25)$ .

#### 4.2.1 Transformació afí

El model de transformació afí consta de 6 paràmetres de moviment i pot representar escalats, translacions i deformacions en algun dels eixos de la imatge. Aquesta transformació ve descrita per les següents equacions:

$$f(x, y) = m[0]x + m[1]y + m[2] \quad (4.2)$$

$$g(x, y) = m[3]x + m[4]y + m[5] \quad (4.3)$$

Aquest model té sis graus de llibertat, i per tant seran necessaris 3 punts per tal de poder resoldre el sistema d'equacions i trobar els paràmetres de moviment  $\vec{m}$  entre dos frames consecutius. Els paràmetres  $m[0]$  i  $m[4]$  controlen l'escalat de la imatge. Els paràmetres  $m[2]$  i  $m[5]$  controlen la translació, igual que en el model translacional, i els paràmetres  $m[1]$  i  $m[3]$  controlen les deformacions en l'eix  $x$  i  $y$  respectivament.

En la figura 4.5 podem observar els tipus de transformacions que es poden obtenir mitjançant el model afí. Amb els paràmetres  $m[2]$  i  $m[5]$  de les equacions (4.2) i (4.3) obtenim la translació (b) de la imatge en la direcció  $x$  i  $y$  respectivament. Modificant els paràmetres  $m[0]$  i  $m[4]$  obtenim el canvi d'escala (c). I amb  $m[1]$  i  $m[3]$  obtenim les deformacions horitzontal (d) vertical (e) respectivament.

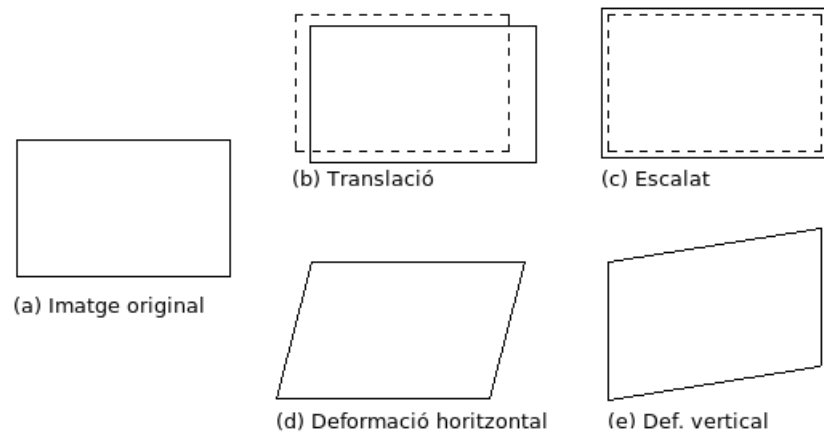
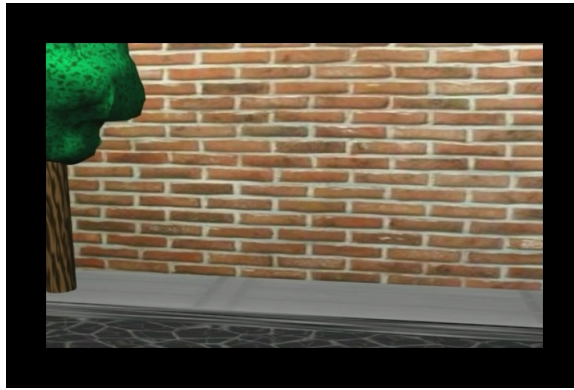
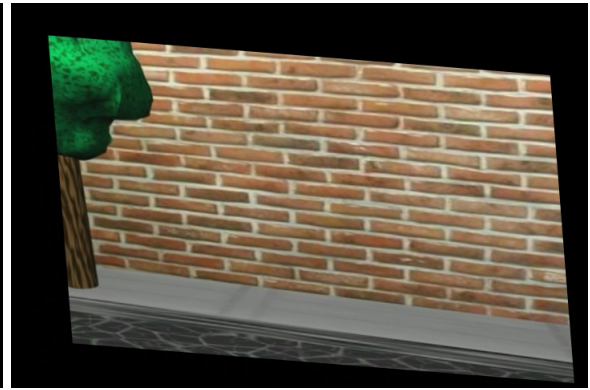


Figura 4.5: (a) Imatge original; (b) Translació, s'aconsegueix modificant els paràmetres  $m[2]$  i  $m[5]$ ; (c) Escalat, modificació dels paràmetres  $m[0]$  i  $m[4]$ ; (d) i (e) Deformació horitzontal i vertical controlada pels paràmetres  $m[1]$  i  $m[3]$  respectivament

En la figura 4.6 podem observar l'efecte produït per la transformació afí sobre un fotograma de la seqüència de prova.



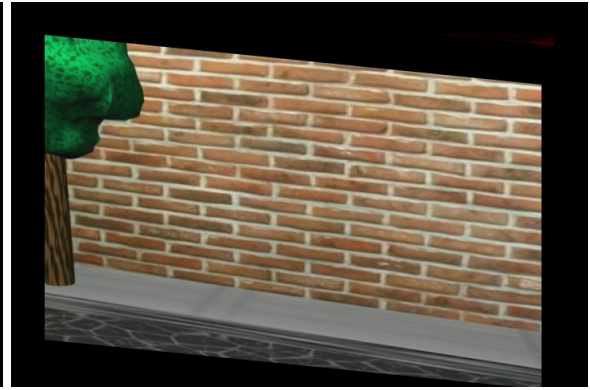
(a) Imatge original



(b) Transformació afí. Els paràmetres de la transformació són (1.01, 0.08, -8, 0.08, 1.01, -14).



(c) Transformació afí sobre l'eix X. Deformació horitzontal



(d) Transformació afí sobre l'eix Y. Deformació vertical

Figura 4.6: Efecte de la transformació afí sobre la imatge

#### 4.2.2 Transformació Perspectiva

La transformació perspectiva, a diferència de les anteriors, no és una aproximació polinòmica de la funció de transformació. Amb les transformacions vistes fins ara les imatges obtingudes un cop aplicada la transformació sempre són paral·lelograms, és a dir, tenen els costats paral·lels dos a dos. Aquesta transformació permet modificar la imatge sense aquesta restricció. Les equacions de la transformació perspectiva són les següents:

$$f(x, y) = \frac{m[0]x + m[1]y + m[2]}{m[6]x + m[7]y + 1}$$

$$g(x, y) = \frac{m[3]x + m[4]y + m[5]}{m[6]x + m[7]y + 1}$$

Com el seu nom indica, a més de totes les transformacions contemplades en els models anteriors, aquesta transformació té en compte la projecció de perspectiva com es pot observar a la figura 4.7.

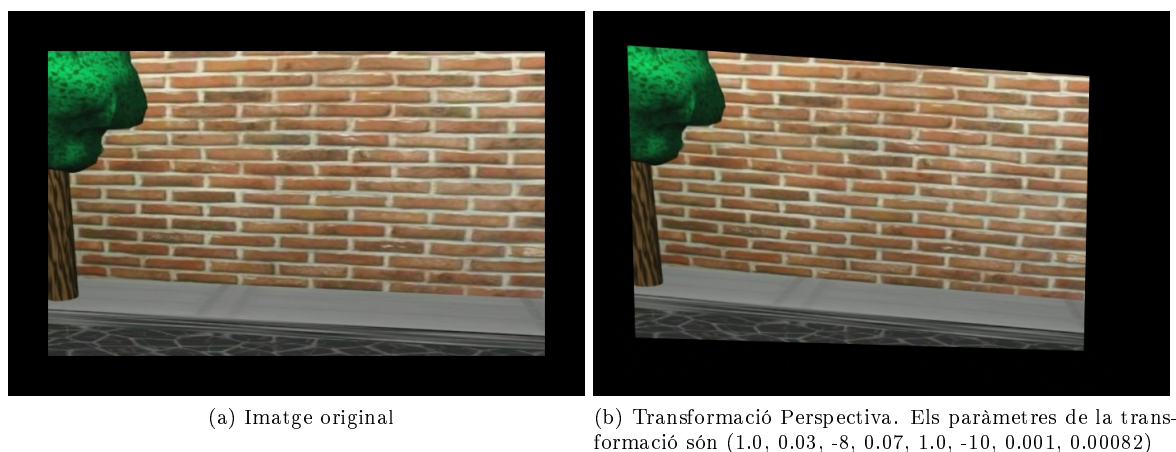


Figura 4.7: Transformació perspectiva

Com es pot observar en les equacions del model de transformació, la transformació perspectiva té 8 graus de llibertat. Això significa que serà necessari establir la correspondència com a mínim entre 4 punts per solucionar el model i obtenir els paràmetres de moviment. Recordem que els punts són bidimensionals i per tant cada punt ens aporta una solució per cada equació del model de transformació.

### 4.2.3 Elecció del model de transformació

Per tal d'elegir el model adequat es necessari tenir en compte el tipus de seqüències de vídeo que es volen processar. Si elegim un model amb més paràmetres dels necessaris, el sistema continuarà funcionant correctament però la velocitat de processat es veurà reduïda considerablement. A més a més, el soroll de l'estimació també serà més elevat com més nombre de paràmetres s'hagin d'estimar. La qualitat de la seqüència de sortida es pot veure afectada negativament per aquest soroll d'estimació. Així, haurem de trobar el model que descrigui la transformació de les imatges amb els menors graus de llibertat possibles. Per contra si el model elegit no permet descriure el moviment de la càmera, no aconseguirem eliminar les vibracions de la seqüència de vídeo. Si per exemple intentem compensar una seqüència amb una transformació afí quan les imatges haurien de ser compensades amb transformació perspectiva, no es podrà aconseguir compensar les imatges i per tant la seqüència de sortida no serà la desitjada. Per tant, l'elecció del model de moviment és un punt d'especial interès ja que d'ell depenen directament els resultats.

En principi sembla lògic afirmar que amb la transformació afí en tindrem suficient per estimar el moviment de la càmera en seqüències de vídeo vigilància, degut a que el moviment serà principalment desplaçament de la càmera i en algun cas podríem tenir un canvi d'escala degut a una vibració. En la figura 4.8 podem observar una imatge panoràmica creada a partir de diversos frames d'una seqüència de vídeo. Es pot observar gràcies a aquesta imatge panoràmica que la transformació afí és suficient per representar les transformacions que apareixen degut al moviment de la càmera.

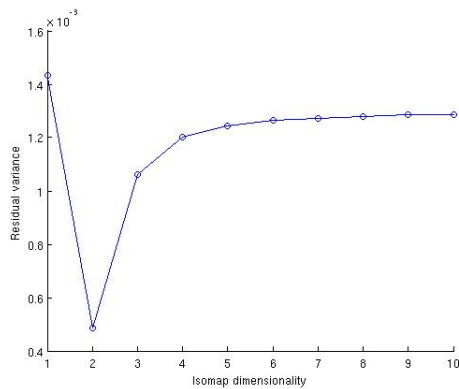
De totes formes, convé assegurar-nos abans de continuar. Per això realitzarem una estimació dels paràmetres utilitzant un model de transformació perspectiva i aplicarem a les mostres



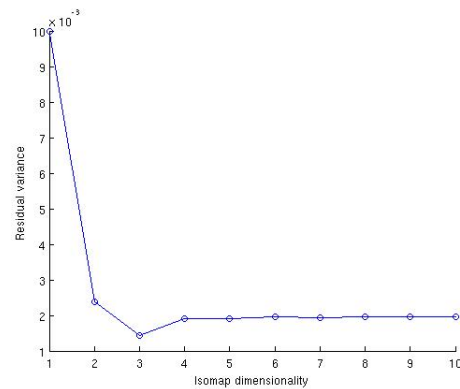
Figura 4.8: Imatge panoràmica creada a partir de diversos frames d'una seqüència de vídeo.

obtingudes l'algorisme descrit en [Tene00]. Aquest algorisme ens mostra la dimensió de les dades d'entrada mitjançant una funció en Matlab que els seus autors han anomenat Isomap.

La funció Isomap ens indica la variància residual en funció de la dimensionalitat del joc de dades. La dimensió intrínseca és el punt on la variància deixa de créixer de forma considerable. Per poder avaluar aquesta funció utilitzarem una seqüència de prova de 500 frames en que la càmera realitza un escombrat horitzontal de l'escena i a més s'han superposat dues vibracions. La primera vibració és en la mateixa direcció de desplaçament de la càmera, és a dir, perpendicular a l'eix focal, i es troba entre els frames 50 i 75, la segona vibració és en direcció longitudinal a l'eix focal i es troba entre els frames 225 i 250. En la figura 4.9 podem veure el gràfic obtingut mitjançant la funció Isomap implementada amb Matlab. Podem observar en (a) que la dimensió intrínseca per al total de frames és de 2. Aquesta informació ens indica que amb una transformació de translació en tindríem suficient per representar matemàticament el moviment de la càmera. Si observem la figura (b) la dimensió canvia a 3. Segons aquesta figura necessitaríem com a mínim 3 paràmetres, i per tant el model de transformació afí per representar el moviment de la càmera. Aquesta diferència és deguda a que en la majoria dels 500 frames de la seqüència el moviment de la càmera és només translacional, mentre que entre el frame 200 i 300 està present la segona vibració, que té un efecte de canvi d'escala sobre la imatge i per tant requereix d'un paràmetre extra. Si elegíssim el model translacional el moviment estaria ben estimat per a la major part de la seqüència però en alguns tipus de vibracions, que és on realment ens importa l'estimació, aquesta seria errònia i per tant no podríem eliminar-la.



(a) Isomap per una seqüència de 500 frames



(b) Isomap per als frames entre el 200 i 300 de la mateixa seqüència

Figura 4.9: Determinació de la dimensió del model de transformació amb l'ajuda de la funció Isomap. En (a) observem que la dimensió del joc de dades és 2 degut a que la major part del moviment d'aquesta seqüència és translacional. Si només agafem una part de la seqüència (b) que conté una vibració en l'eix focal de la càmera la dimensió del joc de dades augmenta a 3.

En la figura 4.10 podem observar una comparació de les estimacions del paràmetre  $m[2]$  de la transformació afí i perspectiva (és el que indica la translació en x) per a la seqüència abans esmentada. S'ha escollit el paràmetre  $m[2]$  degut a que la major part del moviment de la seqüència es troba en el desplaçament al llarg de l'eix X. S'observa que la diferència de les estimacions entre aquests dos models és mínima, tal i com esperàvem, ja que segons la funció Isomap la transformació afí és suficient per estimar el moviment de la càmera en aquestes seqüències.

A la vista d'aquests resultats el model que utilitzarem per eliminar les vibracions serà el model afí. Aquest ens proporcionarà una estimació suficientment bona.

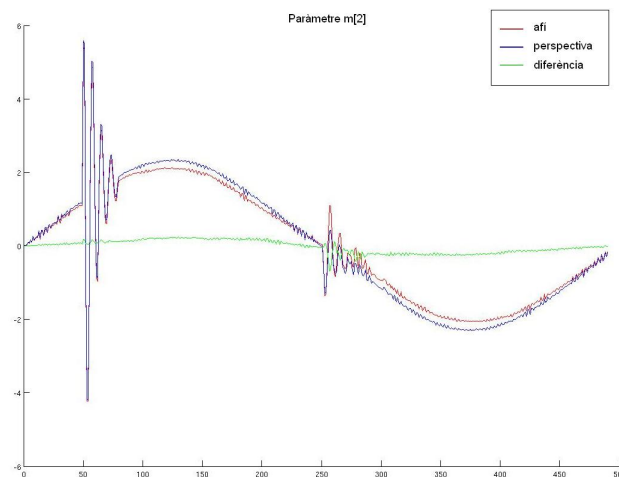


Figura 4.10: Comparació de l'estimació del paràmetre  $m[2]$  entre el model de transformació afí i perspectiva. En vermell tenim l'estimació utilitzant el model afí, en blau l'estimació utilitzant el model de transformació perspectiva i en verd la diferència entre totes dues.

### 4.3 Estimació dels paràmetres de moviment

Un cop s'ha decidit el model de transformació que s'ha d'aplicar per compensar la seqüència de vídeo s'ha de realitzar l'estimació dels paràmetres d'aquest model. Els paràmetres del model els anomenarem paràmetres de moviment. En aquest punt descriurem diferents mètodes per calcular l'estimació dels paràmetres de moviment i discutirem quin és el més adient en cada cas.

L'estimació dels paràmetres de moviment és un dels punts més importants de l'aplicació que s'està desenvolupant. Mitjançant aquesta estimació decidirem si existeix una vibració i aplicarem la corresponent compensació per tal d'eliminar-la. En la figura 4.11 podem observar l'algorisme de manera gràfica.

Com podem observar l'estimació de moviment es realitza entre el frame  $t_i$  i  $t_{i-1}$ , un cop establerts els paràmetres  $\vec{m}$  se separa la vibració del moviment de la càmera i finalment es compensa la vibració.

Existeixen diverses formes de realitzar l'estimació de moviment. Com ja s'ha comentat en 2.2 en el grup de processat d'imatge s'han desenvolupat unes llibreries de software anomenades

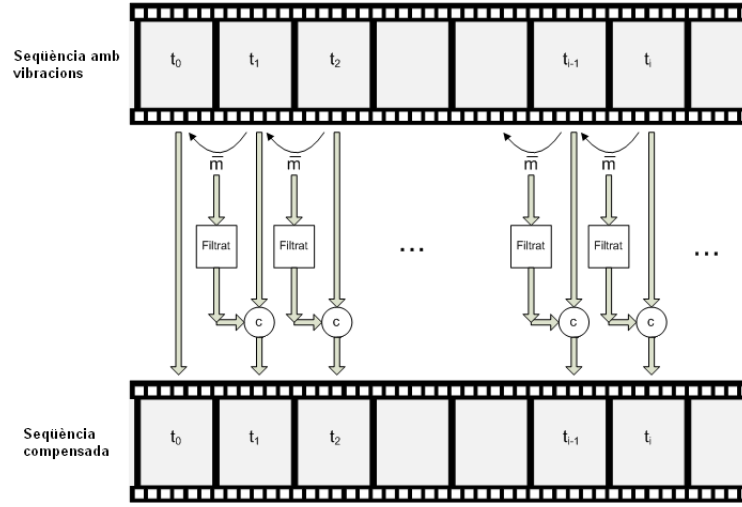


Figura 4.11: Algoritme per a l'eliminació de les vibracions

*Softimage* que permeten treballar amb imatges. Entre altres funcionalitats les que ens interessin en aquest moment són les d'estimació de moviment. La tècnica emprada per realitzar l'estimació de moviment era per descens de gradient utilitzant l'algoritme de Gauss-Newton per optimitzar l'estimació. En el punt 4.3.1 trobem la base d'aquest algoritme. En aquesta aplicació tindrem en compte aquest mètode d'estimació dels paràmetres de moviment ja que funcionava perfectament i era capaç d'eliminar completament les vibracions en seqüències estàtiques.

De totes formes l'eficiència d'aquest mètode és molt baixa ja que només permet processar seqüències a 0.2 fps. Per tal d'assolir l'objectiu descrit en 1.2.2 a la pàgina 6 realitzarem una implementació de l'estimació mitjançant la correlació de les regions veïnes dels punts d'interès trobats mitjançant el detector de Trajkovic-Headley modificat i que s'ha explicat en 3.3 a la pàgina 22.

### 4.3.1 Algoritme de Gauss-Newton

Per tal de realitzar l'estimació dels paràmetres de moviment s'utilitza l'algoritme de Gauss-Newton (GNA) que s'ha introduït en la secció 2.2 a la pàgina 12. Es tracta d'un mètode iteratiu basat en el descens de gradient. La figura 4.12 mostra com un mètode de descens de gradient és capaç de trobar el mínim d'una funció recorrent el camí més curt que és la direcció de major variació de la funció, és a dir, allà on la derivada presenta valors més elevats. A continuació s'explicarà de forma detallada l'algoritme i la seva aplicació en l'estimació dels paràmetres de moviment.

El problema que ens ocupa tracta de trobar els paràmetres de moviment que descriuen la transformació que ha sofert la imatge entre dos frames. Com s'ha comentat en 4.2.3 a la pàgina 31 el model utilitzat per a realitzar l'estimació de moviment és la transformació afí. Per tant, el problema que hem de resoldre és l'estimació dels 6 paràmetres del model afí que descriu la transformació entre dos frames.

$$f(I_i(x, y)) = I_{i-1}(x, y) \quad (4.4)$$

En (4.4) la transformació  $f$  és la corresponent transformació afí que converteix el frame  $i$  de la seqüència en un frame idèntic corresponent a l'instant  $i - 1$ . Així  $f(\bullet)$  vindrà descrita per les següents equacions,

$$\begin{aligned} x_i &= m[0]x_{i-1} + m[1]y_{i-1} + m[2] \\ y_i &= m[3]x_{i-1} + m[4]y_{i-1} + m[5] \end{aligned} \quad (4.5)$$

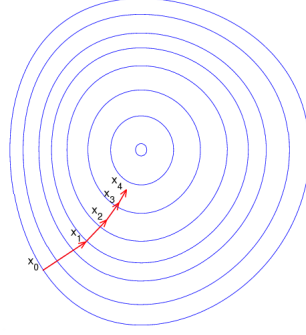


Figura 4.12: Cerca d'un mínim amb el descens de gradient

Aquesta transformació ha de ser vàlida per tots els píxels de la imatge  $I$ .

Definim el vector  $p = (m[0], m[1], m[2], m[3], m[4], m[5])$  de paràmetres de moviment i  $\hat{I}$  l'estimació de la imatge  $I_{i-1}$  feta a partir de la imatge  $I_i$ ,

$$I_{i-1} \approx \hat{I} = f(p)$$

A partir del valor inicial  $p_0$  hem de trobar l'estimació de  $p$  tal que minimitza l'error entre l'estimació i la imatge en l'instant  $i - 1$ ,

$$\epsilon = I_{i-1} - \hat{I}$$

La base de l'algoritme de GN és una aproximació lineal de  $f$  en un entorn al voltant de  $p$ . Per un  $\|\delta_p\|$  petit podem assumir l'aproximació mitjançant una sèrie de Taylor

$$f(p + \delta_p) \approx f(p) + \mathbf{J}\delta_p$$

on  $\mathbf{J}$  és la matriu Jacobiana  $\frac{\partial f(p)}{\partial p}$ . Començant del punt inicial  $p_0$ , el mètode GN produeix una sèrie de vectors  $p_1, p_2, \dots$  que convergeixen a un mínim local de  $\epsilon$ . A cada pas, s'ha de trobar la quantitat  $\delta_p$  que minimitza

$$\|I_{i-1} - f(p + \delta_p)\| \approx \|I_{i-1} - f(p) - \mathbf{J}\delta_p\| = \|\epsilon - \mathbf{J}\delta_p\|$$

El  $\delta_p$  buscat és la solució d'un problema de mínims quadrats i s'assoleix quan  $\mathbf{J}\delta_p - \epsilon$  és ortogonal a l'espai vectorial constituït per les columnes de  $\mathbf{J}$ . Això passa quan  $\mathbf{J}^T(\mathbf{J}\delta_p - \epsilon) = 0$ . Així doncs,  $\delta_p$  es pot calcular com la solució de les equacions normals,

$$\mathbf{J}^T \mathbf{J} \delta_p = \mathbf{J}^T \epsilon$$

La matriu  $\mathbf{J}^T \mathbf{J}$  és una aproximació de la matriu Hessiana i per tant podem reescriure l'equació com

$$\mathbf{H} \delta_p = \mathbf{J}^T \epsilon$$

La solució del problema s'obté calculant la inversa de la matriu hessiana  $\mathbf{H}^{-1}$ , el vector  $\mathbf{J}$  de derivades parcials i l'error de l'estimació  $\epsilon$

$$\delta_p = \mathbf{H}^{-1} \mathbf{J}^T \epsilon$$

Com tot mètode iteratiu s'ha definit un nombre màxim d'iteracions i uns límits en els diferents valors de  $\delta_p$  per determinar el final de la cerca.



Un dels problemes que presenta el mètode de Gauss-Newton a l'hora de cercar el mínim d'una funció és que si l'estimació inicial  $p_0$  està molt allunyada de la solució buscada, es possible que el mètode caigui en un mínim local o la convergència sigui més lenta. Per tal de solucionar aquests problemes s'ha optat per realitzar el mètode utilitzant una piràmide multi nivell. La piràmide es construeix reduint el tamany de les imatges a la meitat per a cada nivell. És a dir, si s'ha de processar una imatge de  $640 \times 480$  es construirà una piràmide de tres nivells. En el primer d'ells les imatges seran de  $160 \times 120$  píxels. En el segon nivell tindrem les imatges de  $320 \times 240$  píxels i en el tercer la imatge amb el tamany original. L'algoritme començarà buscant el mínim a partir de la inicialització  $p_0$  en el primer nivell, imatges petites. Seguidament utilitzarà l'estimació trobada en aquest nivell per inicialitzar l'algoritme en el segon nivell i finalment realitzarà la mateixa operació per al tercer nivell. Aquest mètode ens permetrà augmentar la velocitat de convergència i reduir la probabilitat que la solució trobada sigui un mínim local.

### 4.3.2 Correlació

Per tal d'intentar millorar la velocitat de processat s'ha implementat una estimació dels paràmetres de moviment basada en la correlació. La correlació s'ha implementat mitjançant la FFT i s'ha utilitzat la llibreria `fftw3` [fftw] per a tal fi.

En  $\mathbb{R}^1$  la correlació ens proporciona informació de semblança entre dues senyals. Si tenim dues senyals idèntiques però una retardada respecte l'altra, la correlació ens permet conèixer el retard entre les dues. Aquest mateix concepte el podem aplicar a  $\mathbb{R}^2$ . Quan tractem imatges, cada píxel  $I(x, y)$  es caracteritza per la seva posició en  $x$  i en  $y$ . Mitjançant la correlació bidimensional podem establir el desplaçament que existeix entre dues imatges. En la figura 4.13 es pot observar aquest desplaçament. La correlació entre aquestes dues imatges ens proporciona un màxim en el punt corresponent al desplaçament entre elles (31, 31).



(a) Imatge 1

(b) Imatge 2 desplaçada respecte 1

Figura 4.13: Exemple de desplaçament entre imatges

En general per a seqüències complexes  $x(n)$  i  $y(n)$ , si

$$x(n) \xrightarrow{FFT} X(k)$$

$$y(n) \xrightarrow{FFT} Y(k)$$

llavors,

$$r_{xy}(l) \xrightarrow{FFT} R_{xy}(k) = X(k)Y^*(k)$$

on  $r_{xy}(l)$  és la correlació circular creuada. D'aquesta manera, utilitzant una DFT ràpida o FFT podem calcular ràpidament la correlació entre els dos senyals. Els senyals que estem tractant són imatges de  $N \times N$  píxels. Si calculéssim la correlació mitjançant la DFT el cost computacional seria de  $N^2 \times N^2 = N^4$ , mentre que utilitzant la FFT el cost computacional és de  $N^2 \log_2 N$ .

La correlació  $C(\delta_x, \delta_y)$  entre dues imatges  $A$  i  $B$  de  $M \times N$  píxels es defineix com,

$$C(\delta_x, \delta_y) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} A(x, y) \times B(x + \delta_x, y + \delta_y)$$

La posició  $(\delta_x, \delta_y)$  en que  $C$  sigui màxima determina la translació que millor representa el desplaçament entre les dues imatges.

Igual que en el cas unidimensional, la correlació per a les imatges es pot calcular mitjançant la transformada de Fourier per tal de reduir el cost computacional. D'aquesta manera el càlcul de la correlació entre dues imatges del mateix tamany és,

$$C(\delta_x, \delta_y) = \mathcal{F}^{-1}(\mathcal{F}(A) \times \mathcal{F}(B)^*)$$

De totes formes, el problema que ens ocupa no requereix el càlcul de la correlació entre dues imatges del mateix tamany. Un cop trobats els punts d'interès utilitzarem una subimatge al voltant d'aquests punts per buscar en el frame següent. La regió de cerca tampoc serà tot el frame, sinó que l'acotarem a una subregió al voltant del punt d'interès aprofitant que la càmera no realitza moviments bruscs, i per tant, el desplaçament d'un píxel entre dos frames consecutius està acotat. En la figura 4.14 es pot observar una representació de la situació explicada on podem observar la subimatge al voltant del punt d'interès i la regió de cerca en el frame de referència.

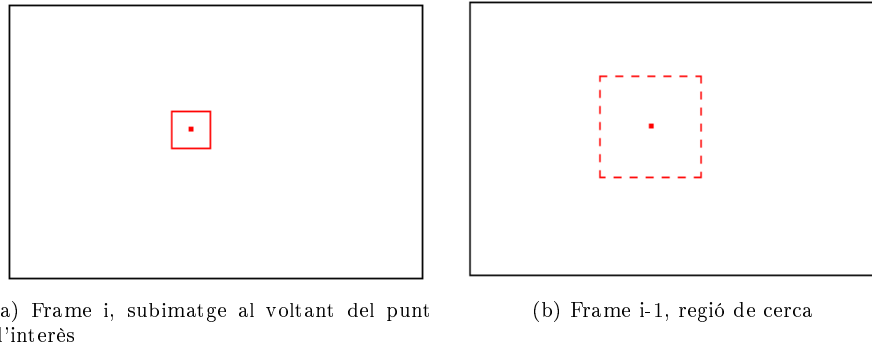


Figura 4.14: Exemple de subimatge i regió de cerca al voltant d'un punt d'interès.

En aquest cas les dues imatges tenen tamany diferents. Una tècnica utilitzada consisteix en fer la subimatge del mateix tamany que la regió de cerca afegint zeros. Aquesta tècnica, però, no ens proporciona informació útil. Un estudi realitzat per Werner Van Belle en [Werner07] demostra que per solucionar aquest problema primer s'han de normalitzar les imatges restant la mitja i posteriorment normalitzar la desviació estàndard. Per realitzar aquest processat presenta un algoritme eficient basat en una finestra de promitjat.

Finalment per millorar l'estimació, l'algoritme de Werner utilitza la correlació com un mapa per buscar la suma de diferències absolutes (SAD). La regió de cerca es divideix en parts iguals

i per cada part es col·loca la subimatge en el punt de màxima correlació i es calcula la SAD. Quan s'ha fet per a totes les parts, s'estableix com a desplaçament el punt que ha donat una SAD menor, tal com es mostra en la figura 4.15,

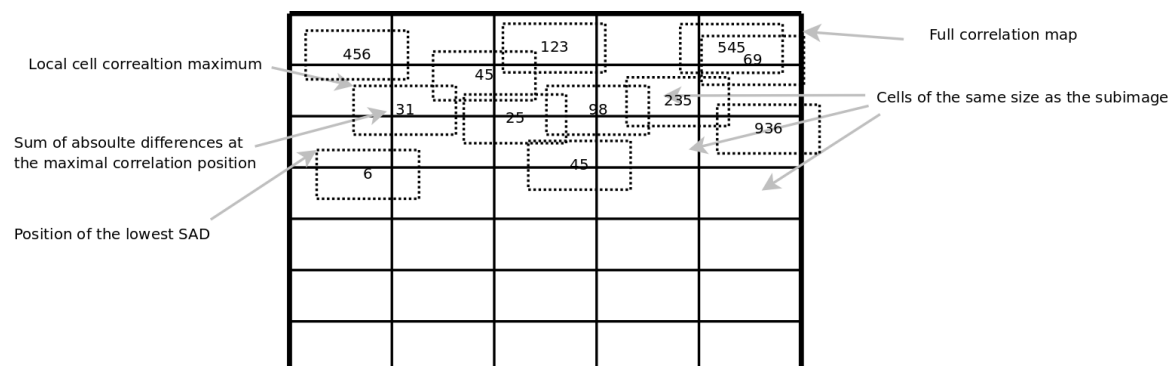


Figura 4.15: S'utilitza la correlació com un mapa per calcular la SAD i establir el desplaçament entre frames consecutius.

Per tal de comprovar el funcionament d'aquest algoritme s'ha realitzat una cerca en un frame de la seqüència de prova. Hem extret una subimatge de 32x32 píxels del frame i l'hem buscat en una regió de 64x64 píxels. Els resultats obtinguts es poden observar en la figura 4.16.

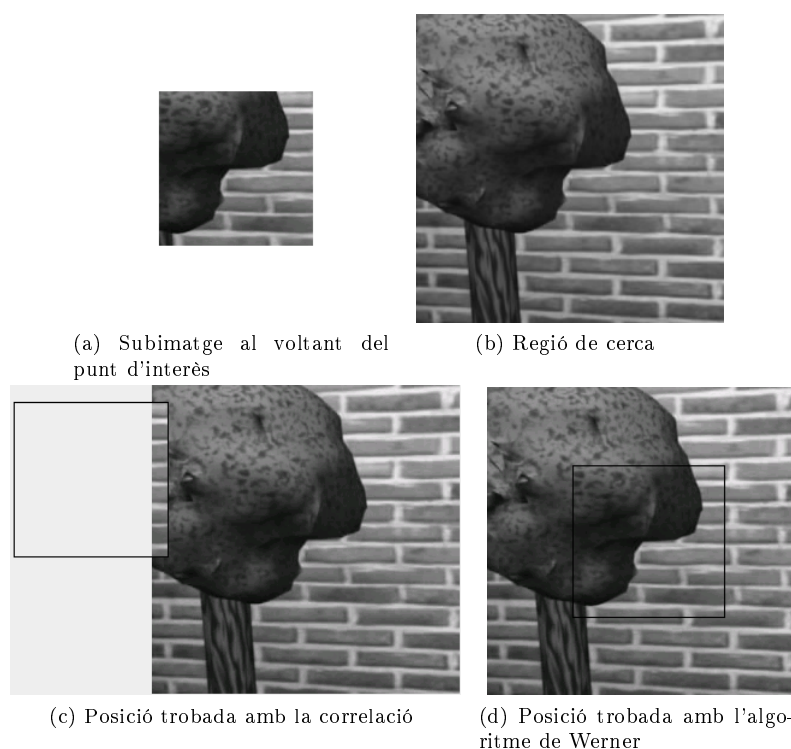


Figura 4.16: Comportament de l'algoritme de Werner per un frame de la seqüència de prova.

### Resolució del model de transformació

Un cop trobada la correspondència entre els punts d'interès trobats entre dos frames consecutius, hem de resoldre el sistema d'equacions corresponent al model de transformació escollit. En 4.2.3

s'ha justificat l'elecció del model afi. Aquest model presenta 6 paràmetres de moviment, i per tant seran necessaris, com a mínim, 3 punts per poder calcular-los. De totes formes amb 3 únics punts no podem establir el moviment global de tota la imatge. Per trobar els paràmetres de moviment globals, primer buscarem els paràmetres locals mitjançant agrupacions de 3 punts i tot seguit calcularem la mitja dels paràmetres obtinguts.

Per calcular els paràmetres globals hem d'agrupar els punts obtinguts en el bloc anterior, cerca de punts d'interès, de tres en tres. Aquesta agrupació l'hem de realitzar de forma que els punts no es trobin alineats, ja que sinó, tindríem un sistema de 6 equacions amb 6 graus de llibertat, però les equacions serien linealment dependents i no el podríem resoldre. Gràcies a la segona modificació introduïda en el detector de punts d'interès descrita en 3.3 a la pàgina 22 podem realitzar les agrupacions de manera molt senzilla i assegurant que els punts mai estaran alineats. Aquesta agrupació la farem dividint la imatge en 3 zones tal com es mostra en la figura 4.17. Les triades de punts les construirem agafant un punt de cada zona tal i com mostren les fletxes en aquesta figura, començant per les regions marcades amb una circumferència i acabant per les regions marcades amb un cercle.

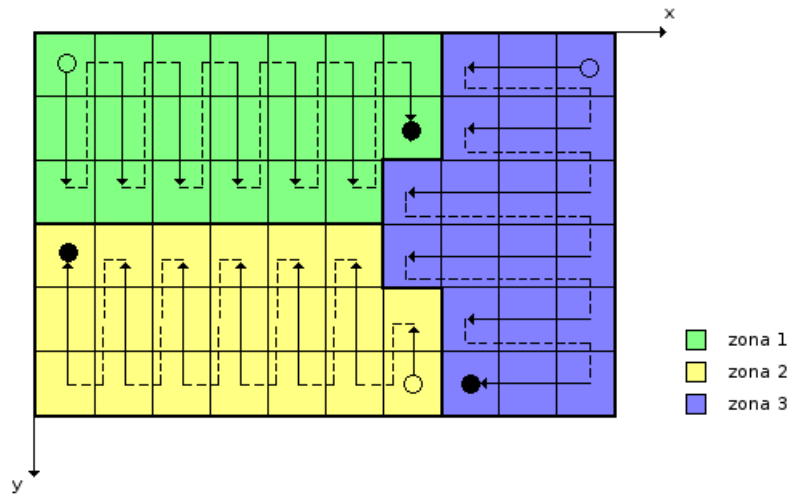


Figura 4.17: Divisió de la imatge en regions i separació de les regions en 3 zones. Elegim un punt de cada zona per confeccionar les triades per resoldre el model de transformació. La primera triada estarà formada pel punt trobat en les regions marcades amb una circumferència i seguirem creant les triades seleccionant els punts tal com indiquen les fletxes.

El detector de Trajkovic-Headley amb la segona modificació ens ajuda molt a l'hora de crear les triades de punts. Aquest detector dividia la imatge en diferents regions i buscava per cada regió el punt on hi ha una variació màxima d'intensitat maximitzant una funció d'intensitat o SAD entre el punt en qüestió i els seus veïns. Aquest funcionament ens permet assegurar que per a cada regió de la imatge tindrem un punt. A més els punts estaran ordenats de forma que només amb l'índex del vector de punts podem saber a quina regió es troba sense haver de fer més càlculs.

A partir d'aquests tres punts en el frame de referència,  $(x0[n], y0[n])$ , i els seus corresponents en el frame actual,  $(x1[n], y1[n])$ , resoldrem el sistema d'equacions lineal 4.6 de la transformació afí que podem reescriure com,

$$x1[n] = m[0]x0[n] + m[1]y0[n] + m[2]$$

$$y1[n] = m[3]x0[n] + m[4]y0[n] + m[5]$$

El sistema d'equacions el podem escriure amb forma vectorial,

$$\bar{b} = A \times \bar{m} \quad (4.6)$$

on els vectors  $\bar{m}$ ,  $\bar{b}$  i la matriu  $A$  estan definides de la manera següent:

$$\bar{b} = \begin{bmatrix} x1[0] \\ x1[1] \\ x1[2] \\ y1[0] \\ y1[1] \\ y2[2] \end{bmatrix} \quad \bar{m} = \begin{bmatrix} m[0] \\ m[1] \\ m[2] \\ m[3] \\ m[4] \\ m[5] \end{bmatrix} \quad A = \begin{bmatrix} x0[0] & y0[0] & 1 & 0 & 0 & 0 \\ x0[1] & y0[1] & 1 & 0 & 0 & 0 \\ x0[2] & y0[2] & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x0[0] & y0[0] & 1 \\ 0 & 0 & 0 & x0[1] & y0[1] & 1 \\ 0 & 0 & 0 & x0[2] & y0[2] & 1 \end{bmatrix}$$

Resolent (4.6) obtenim els paràmetres de moviment corresponent a la transformació afí.

$$\bar{m} = A^{-1} \times \bar{b} \quad (4.7)$$

Per tant, per trobar els paràmetres de la transformació haurem de resoldre el sistema d'equacions per a cada agrupació de 3 punts. Un cop trobats els paràmetres per a totes les triades de punts de la imatge haurem de realitzar un mitja de tots ells per obtenir els paràmetres de moviment globals de la imatge. Amb això es pretén reduir l'error d'estimació. Una altra tècnica emprada per a la reducció de l'error d'estimació consisteix en calcular la mitja i desviació estàndard dels paràmetres i descartar aquells que no es trobin a  $\bar{m} \pm \sigma$ . Amb els paràmetres que compleixen aquesta condició realitzarem la mitjana final.

## Capítol 5

# Eliminació de vibracions i compensació

En els capítols anteriors hem vist una descripció global del sistema per eliminar les vibracions de seqüències de vídeo i aconseguir, d'aquesta forma, les seqüències compensades o lliures de vibracions. Posteriorment hem analitzat tots els mòduls que conformen el sistema per separat. En el capítol 3 s'ha estudiat la detecció de punts d'interès d'una imatge per tal de processar-la utilitzant els mínims punts possibles que descriguin el comportament global. En el capítol 4 s'han estudiat diferents models de transformació i s'han presentat diverses formes d'estimar els paràmetres de les esmentades transformacions. En aquest capítol s'explicarà com el software compensa les seqüències de vídeo utilitzant aquests blocs i obté el vídeo lliure de vibracions. En la figura 5.1 es pot observar la ubicació d'aquest capítol dintre del diagrama de blocs de l'aplicació.

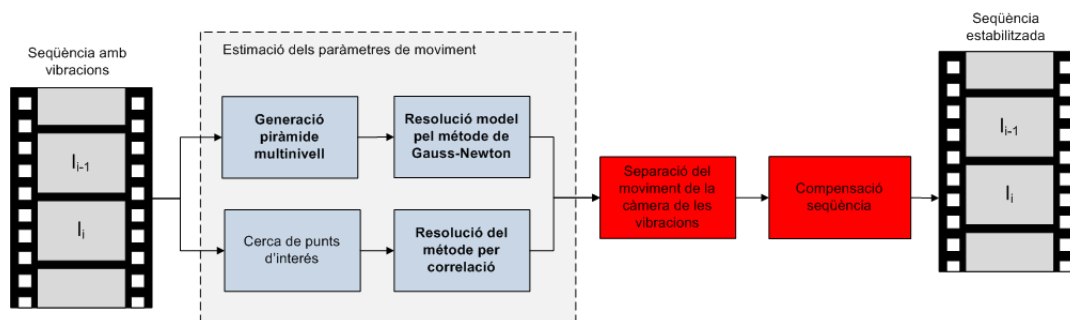


Figura 5.1: Diagrama de blocs de l'aplicació. En vermell els blocs corresponents a l'eliminació de vibracions i obtenció de seqüències compensades.

S'han estudiat tres casos en els que es pot aplicar el sistema d'eliminació de vibracions. Aquests casos estan basats en el comportament de la càmera que enregistra les seqüències de vídeo. Així doncs, distingirem:

- Càmera sense moviment
- Càmera amb moviment conegut
- Càmera amb moviment desconegut a priori

## 5.1 Càmera sense moviment

El cas en que la càmera no presenta moviment serà el més fàcil de resoldre. En aquest cas simplement haurem d'anar llegit successivament els frames de la seqüència de vídeo d'entrada, fer una estimació del moviment entre el frame llegit i el frame origen, que és el primer frame de la seqüència i realitzar la compensació del frame llegit. En aquest cas el moviment detectat en la seqüència de vídeo és produït íntegrament per les vibracions ja que la càmera no s'hauria de moure. Per tant el bloc "Separació de moviment de la càmera de les vibracions" de la figura 5.1 decideix sempre que el moviment és degut a la vibració.

Quan la càmera que enregistra la seqüència de vídeo no presenta moviment podem agafar el primer frame com a referència. Aquest serà el que utilitzarem durant tota l'execució per realitzar la compensació de moviment. Tal com es mostra en la rama en color blau de la figura 5.6 a la pàgina 46, per compensar el moviment simplement anirem llegint els diferents frames de la seqüència (la variable *time* controla el frame actual de la seqüència de vídeo) i fent l'estimació de moviment respecte el fotograma de referència. Si els paràmetres de moviment estimats difereixen dels paràmetres de la càmera en repòs s'haurà de realitzar la compensació de moviment. Per tal propòsit s'aplicarà la transformació inversa al fotograma llegit i d'aquesta manera aconseguirem que la seqüència de vídeo de sortida quedi estabilitzada, és a dir, lliure de moviments indesitjats. Repetim el mateix procés fins que arribem al final de la seqüència en cas que estem processant una seqüència enregistrada, o indefinidament si es tracta d'una seqüència capturada en temps real.

La figura 5.2 mostra un esquema de la compensació sense moviment.

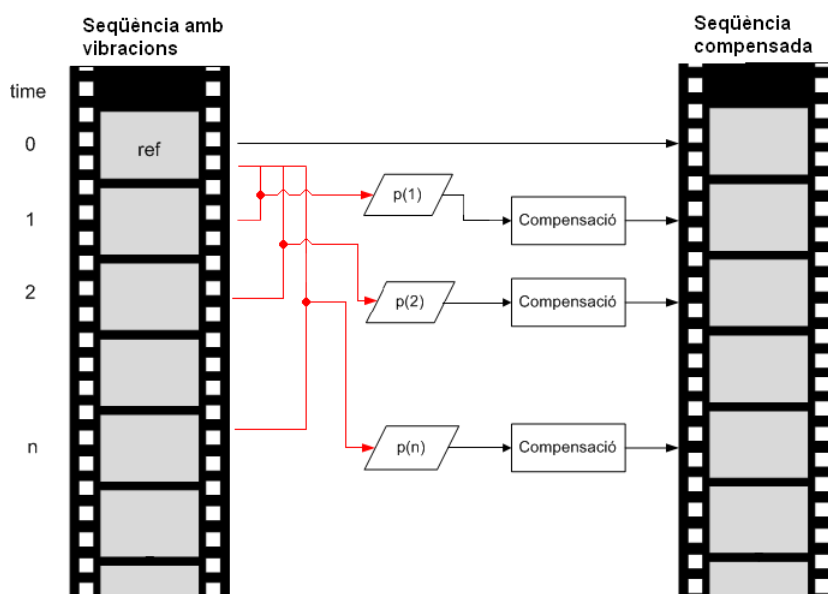


Figura 5.2: Esquema de compensació sense moviment

## 5.2 Càmera amb moviment conegut

Una situació habitual en la que ens podem trobar és la que la càmera realitza un moviment conegut. Aquest es pot tractar d'un escombrat d'una zona determinada, per exemple. En

aquest cas resulta lògic pensar que podem conèixer els paràmetres de moviment de la càmera a priori. A partir del programa que gestionarà el servomotor que fa moure la càmera i el calibratge d'aquesta podem aplicar una transformació adient per tal de trobar els paràmetres que descriuen el seu moviment. Aquesta transformació no entra dintre dels objectius d'aquest projecte i per això es deixa com a treball futur en la secció 7.2.

En el cas en que la càmera presenta un moviment conegut, la compensació de moviment és bastant pareguda al cas en que no hi ha moviment. La principal diferència radica en el fet que, en aquest cas, el frame de referència canvia en cada iteració. Per realitzar l'estimació de moviment agafem com a fotograma de referència l'anterior. En la figura 5.3 es pot observar com el programa ha de realitzar la compensació dels paràmetres de moviment. En aquest diagrama les fletxes representen el paràmetres de moviment i els punts representen la posició d'un determinat píxel dintre de fotogrames correlatius prenent sempre el mateix sistema de coordenades. El color negre representa els paràmetres reals de la càmera i coneguts. El color vermell és l'estimació de moviment i el color verd representa la compensació que s'ha d'aplicar per estabilitzar la seqüència. El quadrat exterior delimita l'àrea del fotograma.

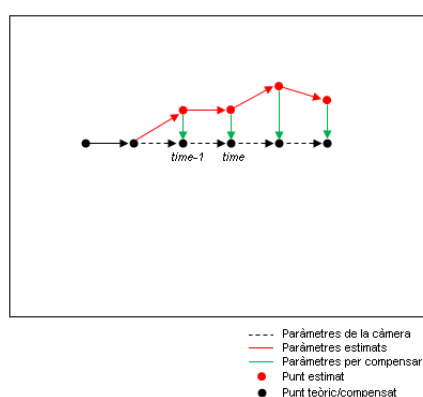


Figura 5.3: Compensació dels paràmetres d'una seqüència amb moviment conegut

El moviment de la càmera en el frame *time* s'estima agafant com a referència el frame *time - 1*. A cada iteració s'ha d'anar acumulant el moviment calculat amb el moviment anterior degut a que el càlcul es fa respecte el frame anterior. Si ens fixem en el quart punt negre de la figura 5.3 observem que, malgrat que els paràmetres estimats són iguals que els reals, la compensació no és nul·la sinó que s'acumula la del fotograma anterior. En la figura 5.4 a la pàgina següent es pot observar un esquema de la compensació quan la càmera presenta un moviment. Aquest esquema és vàlid tant quan el moviment és conegut com desconegut a priori, la diferència entre aquestes dues situacions radica en la forma de calcular els paràmetres per compensar la seqüència de vídeo.

En aquest cas els paràmetres de la càmera són coneguts en cada iteració. Per tal de saber si ha hagut una vibració només hem de comparar els paràmetres estimats amb els paràmetres reals i corregir les diferències entre aquests. Ara els paràmetres de referència no seran els d'una càmera estàtica sinó que l'aplicació els haurà de llegir d'algun recurs extern com podria ser un fitxer.



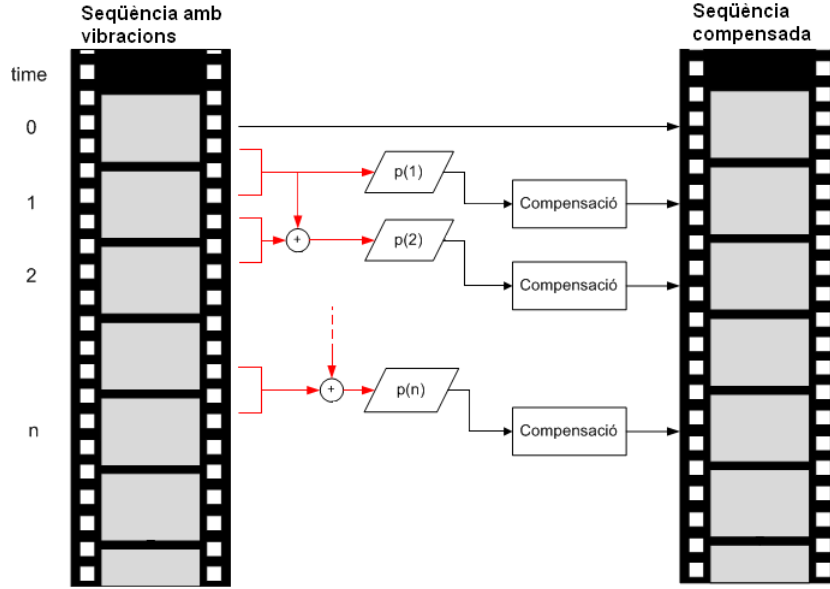


Figura 5.4: Esquema de compensació quan la càmera presenta moviment

### 5.3 Càmera amb moviment desconegut a priori

Aquest cas és el més complicat de resoldre ja que els paràmetres reals de la càmera s'han de calcular a partir de l'estimació de moviment. En principi no suposem cap moviment predeterminat de la càmera però és totalment lògic assumir en una aplicació de videovigilància que la càmera realitzarà l'escombrat d'una determinada regió de l'espai. Separar el moviment de les vibracions serà possible ja que el moviment de la càmera el podem modelar com un moviment sinusoidal de baixa freqüència (5.1) mentre que les vibracions tenen components freqüencials més elevats. Cal remarcar, però, que el software no utilitza cap premissa per al moviment de la càmera.

Assumirem que les vibracions es comporten com un soroll impulsional, que podem modelar com un moviment sinusoidal finit, limitat a 3 o 4 períodes, de freqüència molt superior al moviment de la càmera. Les equacions (5.1) i (5.2) descriuen, respectivament, el moviment de la càmera i la vibració que se superposa.

$$m_c = A_c \times \sin(\omega_c t + \varnothing_c) \quad (5.1)$$

$$m_v \approx A_v \times \prod \left( \frac{t}{4T_{v1}} \right) \times e^{\frac{t_0-t}{T_{v2}}} \times \sin(\omega_{v1}t + \varnothing_v) \quad (5.2)$$

el moviment resultant ( $m_r$ ), serà doncs

$$m_r = A_c \times \sin(\omega_c t + \varnothing_c) + A_v \times \prod \left( \frac{t}{4T_{v1}} \right) \times e^{\frac{t_0-t}{T_{v2}}} \sin(\omega_{v1}t + \varnothing_v)$$

Aquestes vibracions podrien estar causades per diversos factors com podria ser, per exemple, cops en el suport de la càmera o ràfegues de vent en el cas que tinguem càmeres exteriors. En la figura 5.5 es pot observar una realització del moviment descrit. L'eix x de la gràfica conté els frames de la seqüència de vídeo i en l'eix de les ordenades trobem representat el desplaçament de la càmera entre el frame  $x$  i  $x - 1$ .

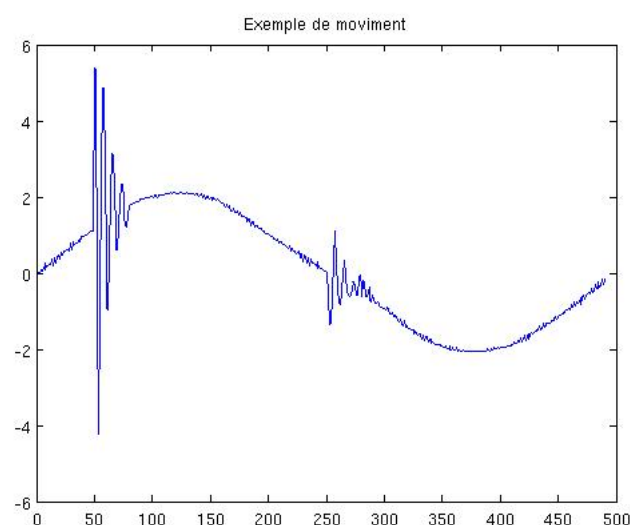


Figura 5.5: Exemple de moviment d'una càmera. Diferenciem l'escombrat de la càmera (moviment sinusoïdal) de la vibració (soroll impulsional)

Un cop estimat, s'ha de separar el moviment de la càmera de la vibració mitjançant un filtrat. Aquest filtrat ens proporcionarà els paràmetres reals de la càmera. Un cop calculats aquests paràmetres podem començar a compensar i generar la seqüència de sortida. En aquest cas el processat de la seqüència de vídeo presentarà un retard si s'utilitza el software en temps real. Aquest retard és degut a la necessitat de conèixer l'estimació d'una sèrie de paràmetres per poder començar a realitzar el filtrat per calcular els paràmetres reals.

Com s'ha comentat anteriorment l'esquema d'aquesta situació és el mateix que per al cas anterior i es pot consultar a la figura 5.4 a la pàgina anterior. Un cop obtinguts els paràmetres reals, l'aplicació realitzarà la mateixa compensació que en el cas de la càmera amb moviment conegut ja que en aquest punt el moviment de la càmera ja no es considera desconegut.

En la figura 5.6 es pot observar l'esquema general de l'aplicació a mode de resum d'aquest capítol. Es pot veure clarament la diferenciació en funció del tipus de seqüència que s'està processant. Durant la fase d'inicialització es llegeix del fitxer de configuració si la càmera que enregistra la seqüència té moviment o no. En cas que en tingui, es llegeixen els paràmetres de referència de la càmera si existeixen. A continuació es comença a processar la seqüència.

En blau veiem el bloc corresponent a l'estimació de moviment per a seqüències estàtiques. Com ja s'ha comentat anteriorment, en aquest cas, s'estima el moviment entre el frame inicial de la seqüència (img0) i el frame que s'està processant (img1). En aquest cas la compensació de moviment és instantània ja que només depèn del frame que s'està processant. Quan s'acaba de processar un frame es comprova si s'ha arribat al final de la seqüència i si no és així es llegeix i processa el següent fotograma.

En verd veiem la compensació de moviment corresponent a seqüències en moviment. S'observa que hi ha una part diferenciada i una part comuna a totes dues. Per a les seqüències amb moviment conegut l'estimació i la compensació és pareguda a la de les seqüències estàtiques. El tret que les diferencia és que, en aquest cas, un cop estimats els paràmetres de moviment entre el frame de referència i el frame que s'està processant s'han de calcular els paràmetres de compensació. Això és degut al fet que els paràmetres de moviment es van acumulant al llarg de la seqüència tal com ja hem vist en aquest capítol. La segona diferència radica en el frame

de referència. En els casos en que la seqüència presenta moviment, el frame de referència va canviant i s'agafa l'anterior en cada iteració. És a dir, quan processem el frame *time* agafem com a referència el fotograma *time-1*.

La compensació de seqüències amb moviment desconegut és un poc més complexa. Igualment que en el cas precedent el frame de referència és l'anterior. Aquí, però, la diferència es deguda a que no tenim els paràmetres de referència i els hem de calcular. Per calcular-los utilitzem un filtrat de mediana dels paràmetres estimats. Per això és necessari que esperem tants fotogrames com l'ordre del filtre per començar a filtrar i a compensar la seqüència de vídeo. En aquest cas, doncs, els primers *nfilter* (ordre del filtre) frames no es podran compensar degut a que no disposem de suficients paràmetres per començar a calcular els paràmetres de referència.

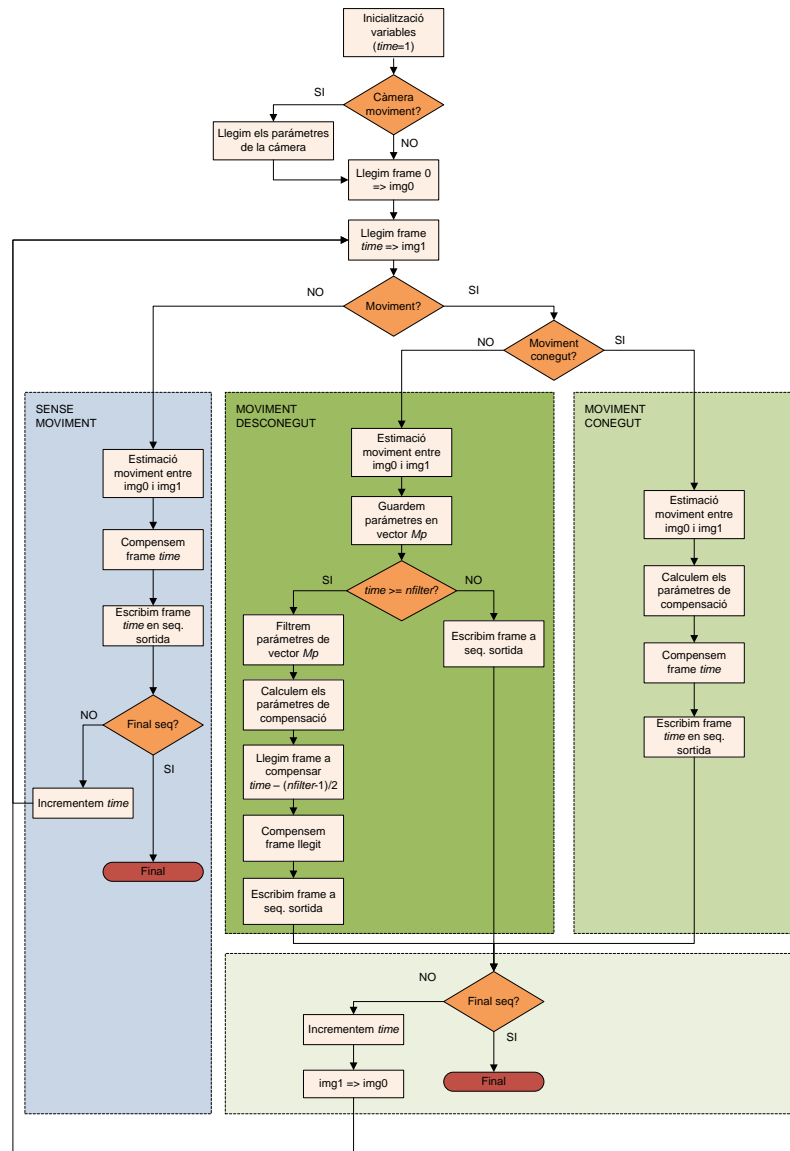


Figura 5.6: Algorisme per a la compensació de moviment

# Capítol 6

## Resultats

Fins ara hem realitzat una descripció de l'aplicació desenvolupada i s'han explicat detalladament tots els blocs que conformen el sistema de compensació de seqüències de vídeo. Per cada bloc funcional de l'aplicació s'ha explicat el seu funcionament i s'han valorat diverses alternatives de disseny per arribar a tenir el sistema complet. En aquest capítol es presentaran els resultats obtinguts amb seqüències de vídeo reals. Per tal propòsit es dividirà el capítol en les tres casuístiques presentades en el capítol 5 a la pàgina 41.

En primer lloc s'analitzaran els resultats obtinguts per a la compensació de seqüències estàtiques i posteriorment s'avaluarà la solució per a seqüències en moviment. En el cas de seqüències en moviment en primer lloc s'avaluarà l'estimació de moviment ja que és comú per als dos casos que s'han tractat: moviment de la càmera conegut i moviment de la càmera desconegut a priori. Un cop presentats els resultats de l'estimació de moviment per a seqüències en moviment es farà la distinció per avaluar els resultats de l'eliminació de vibracions i obtenció de seqüències compensades.

### 6.1 Seqüència de vídeo amb la càmera estàtica

Per avaluar el comportament del software desenvolupat en l'eliminació de vibracions en que la càmera no presenta moviment s'ha utilitzat una seqüència de vídeo corresponent a un aeroport. La càmera rep un cop degut a un operari de l'aeroport i el vídeo presenta una vibració. En la figura 6.1 es poden observar 2 frames d'aquesta seqüència.



Figura 6.1: 2 frames de la seqüència estàtica de l'aeroport

Per tal d'avaluar les prestacions de l'aplicació processarem la seqüència amb els dos mètodes d'estimació de moviment de que disposem: mètode iteratiu per descens de gradient basat en l'algoritme de Gauss-Newton i seguidament mitjançant la correlació.

### 6.1.1 Compensació de vibracions amb descens de gradient

En primer lloc processarem la seqüència amb el mètode de descens de gradient explicat en el punt 4.3.1 a la pàgina 34. Aquest mètode presenta uns resultats molt bons eliminant completament la vibració de la seqüència de vídeo. De totes formes la velocitat de processat és de 0,5 fps. Si ens fixem en la figura 6.2 podem observar l'estimació dels paràmetres de moviment mitjançant aquest mètode. A la dreta tenim els paràmetres estimats de la seqüència de vídeo original. Es pot observar que en els paràmetres  $m(3)$  i  $m(6)$  existeix una vibració en forma de soroll impulsional. Aquesta és la vibració que es produeix quan un operari dona un cop al suport de la càmera. A l'esquerra tenim una estimació dels paràmetres de moviment un cop compensada la seqüència. La vibració s'ha pogut compensar totalment. En la seqüència de vídeo compensada es pot observar com efectivament la vibració ha desaparegut i la seqüència s'ha pogut estabilitzar correctament.

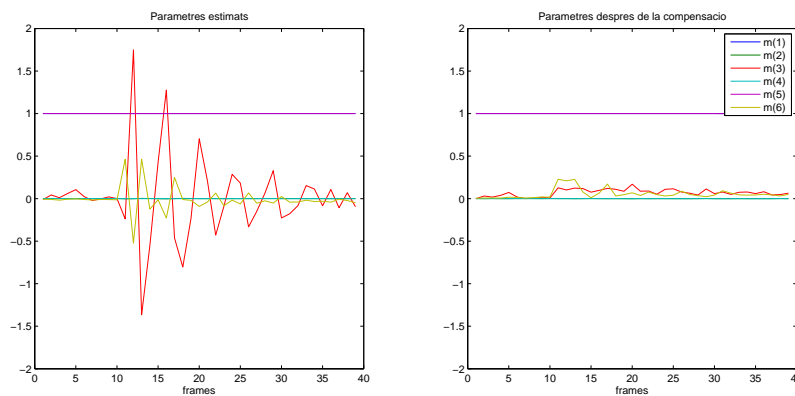


Figura 6.2: Compensació amb el mètode de Gauss-Newton

### 6.1.2 Compensació de vibracions mitjançant la correlació

Per tal d'intentar augmentar la velocitat de processament s'ha fet la compensació utilitzant la correlació. Els resultats no són tan bons com amb el mètode de descens de gradient ja que l'estimador per correlació presenta un nivell de soroll en l'estimació molt més gran. Tot i que la velocitat de processament ha augmentat 4 cops, sent en aquest cas de 2,1fps, està lluny del nostre objectiu de 12fps.

En la figura 6.3 es pot observar l'estimació del moviment sobre la seqüència de l'aeroport. A l'esquerra tenim l'estimació dels paràmetres de moviment de la seqüència original, i a la dreta els paràmetres de la seqüència compensada. S'observa en aquest cas, que la seqüència de sortida presenta vibracions en forma de soroll aleatori produïdes per la pròpia estimació dels vectors de moviment. Observant el vídeo corresponent a aquesta seqüència també s'aprecien les vibracions produïdes per aquesta estimació.

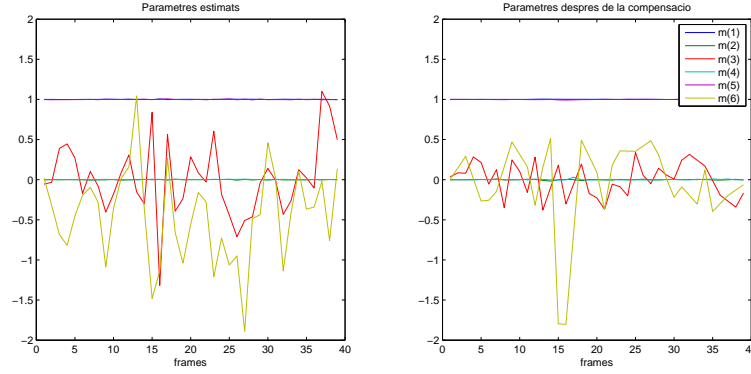


Figura 6.3: Compensació amb el mètode de correlació

S'ha detectat que amb aquesta seqüència en particular funciona millor l'estimador de Trajkovic-Headley amb la primera modificació. Aquesta consisteix en dividir la imatge en tres regions i per cadascuna d'elles busquem els punts mitjançant el detector de Trajkovic-Headley amb la distància mínima. En la figura 6.4 es pot observar els resultats obtinguts de l'estimació dels paràmetres de moviment segons els diferents valors dels paràmetres del detector. En primer lloc podem observar que els paràmetres estimats amb la segona modificació presenten molt més soroll que sense aquesta. Això és degut a que en aquesta seqüència de vídeo els punts d'interès no es troben ben distribuïts per tota la imatge. Hi ha moltes àrees de la imatge que no presenten grans variacions d'intensitat i per això aquestes àrees no haurien d'aportar cap punt al conjunt de punts d'interès. Si utilitzem la segona modificació cada subregió aporta un punt, com podem observar a la figura 6.5. Alguns d'aquest punts no són cap punt d'interès de la imatge i és per això que la correlació ens dona un resultat més dolent.

Com s'ha comentat en el capítol 3, el detector amb distància mínima requereix un ajust dels paràmetres de distància mínima, diàmetre del detector i nivell de comparació. Per tant, abans de processar la seqüència de vídeo definitivament haurem de buscar mitjançant un calibratge quins són els paràmetres més adients per a la seqüència de l'aeroport. Observant les diferents realitzacions presentades de la figura 6.4 es pot establir que els paràmetres que millors resultats donen són els següents:

$$\begin{cases} O=3 \\ threshold = 1500 \\ d_{min} = 10 \end{cases}$$

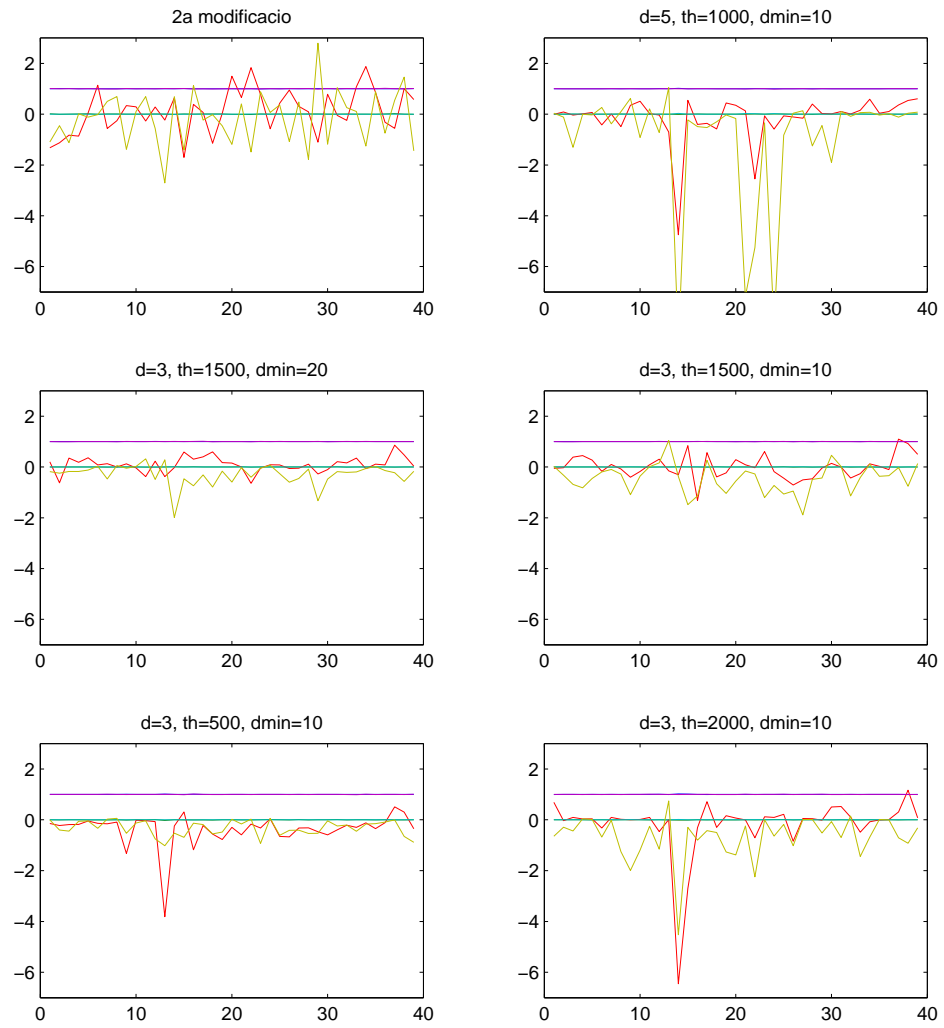


Figura 6.4: Paràmetres de la seqüència compensada variant la configuració del detector

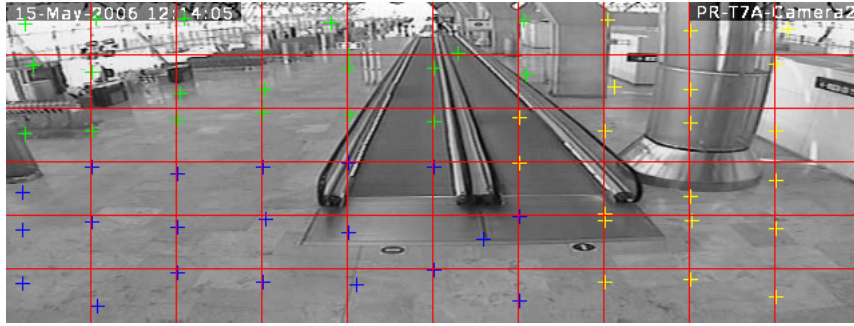
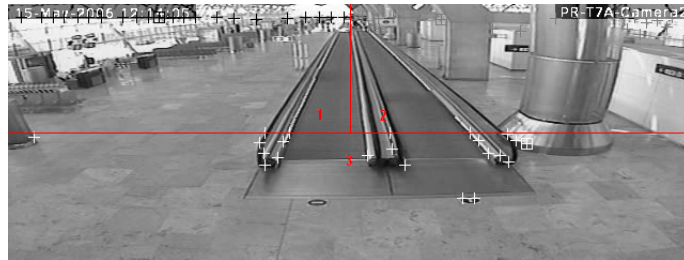
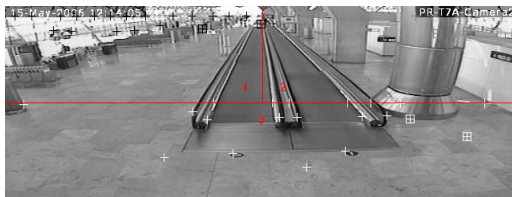


Figura 6.5: Cerca de punts d'interès amb el detector de Trajkovic-Headley amb la segona modificació

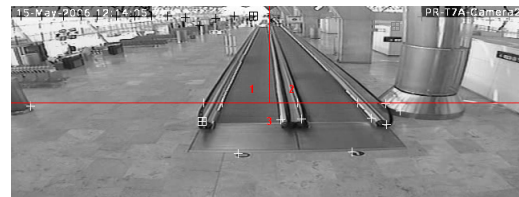
Recordem que la segona modificació del detector de Trajkovic-Headley consistia en dividir la imatge en una graella i buscar el punt amb un valor major de la SAD (Sum of absolute differences). En la figura 6.5 podem veure la graella en que es divideix la imatge i els punts detectats per cada casella de la graella. També s'ha pintat els punts de diferents colors per fer èmfasi a la forma en que es combinen els punts per fer les triangulacions. Per recordar la forma en que s'agrupen els punts es pot consultar la figura 4.17 a la pàgina 39.



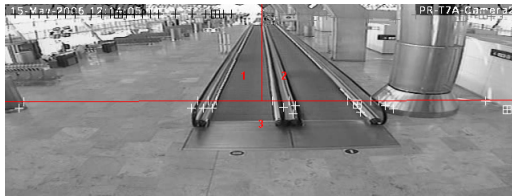
(a)  $\alpha = 3$ ,  $th = 1500$ ,  $d_{min} = 10$



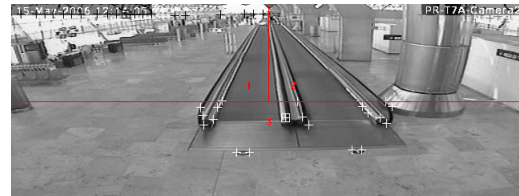
(b)  $\alpha = 5$ ,  $th = 1000$ ,  $d_{min} = 20$



(c)  $\alpha = 3$ ,  $th = 1500$ ,  $d_{min} = 20$



(d)  $\alpha = 3$ ,  $th = 500$ ,  $d_{min} = 10$



(e)  $\alpha = 3$ ,  $th = 2000$ ,  $d_{min} = 10$

Figura 6.6: Punts detectats segons els paràmetres del detector



En la figura 6.6 es pot observar que els punts detectats mitjançant el detector de Trajkovic-Headley amb la 1a modificació estan agrupats sobre certes regions de la imatge que són on es troben els vertaders vèrtexs de la imatge. Recordem que en la primera modificació del detector es dividia la imatge en 3 regions per poder escollir les triades de punts per resoldre el model de transformació afí més fàcilment. En els diferents frames s'han delimitat i numerat les regions. Per resoldre el model de transformació s'agafa un punt de cada regió i posteriorment es fa la mitja dels paràmetres calculats per cada tríada.

Per tal de comprovar el funcionament amb una altra seqüència de vídeo s'ha enregistrat en un dels laboratoris del TSC una seqüència estàtica. En la figura 6.7 es poden observar alguns fotogrames d'aquesta seqüència.

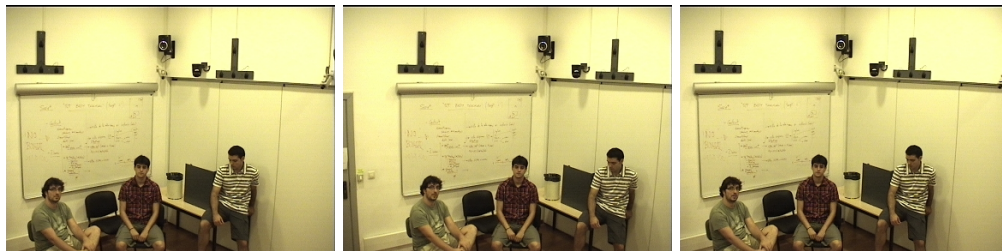


Figura 6.7: Fotogrames de la seqüència estàtica enregistrada en un laboratori del TSC

Per a aquesta seqüència s'ha fet una compensació de moviment utilitzant els dos mètodes descrits: algoritme de Gauss-Newton i correlació. En la figura 6.8 es pot observar els resultats obtinguts mitjançant l'algoritme de Gauss-Newton. En (a) veiem l'estimació dels paràmetres de moviment  $p[3]$  i  $p[6]$ . En aquest gràfic es veu com una vibració ha contaminat la seqüència. En la sub-figura (b) observem l'estimació dels paràmetres després d'haver compensat la seqüència. Malgrat que existeix una vibració residual aquesta és pràcticament imperceptible en la seqüència compensada.

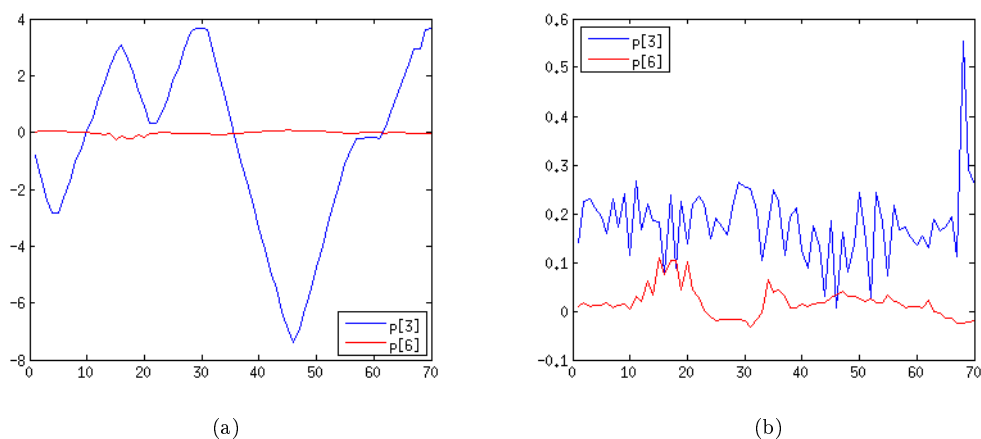


Figura 6.8: Estimació dels paràmetres de moviment de la seqüència estàtica amb el mètode de Gauss-Newton. En (a) trobem l'estimació de moviment dels paràmetres de desplaçament horitzontal  $p[3]$  i vertical  $p[6]$ . En (b) hi ha l'estimació de moviment sobre la seqüència compensada.

L'escena també s'ha intentat compensar utilitzant el mètode de correlació. Per tal de realitzar aquesta estimació s'ha determinat que el detector de punts d'interès que millor funciona per aquesta seqüència és el corresponent a la primera modificació, és a dir, dividir la imatge en 3 zones i aplicar el detector amb distància mínima per a cadascuna d'elles. Els paràmetres que s'han extret del calibratge per aquesta seqüència són els següents:

$$\begin{cases} \emptyset = & 7 \\ threshold = & 2500 \\ d_{min} = & 5 \end{cases}$$

Amb aquests paràmetres s'obtenen els punts que es poden observar en la figura 6.9. Es pot observar que aquesta seqüència presenta grans àrees sense punts d'interès i és per això que funciona millor aquest detector que el corresponent a la segona modificació.

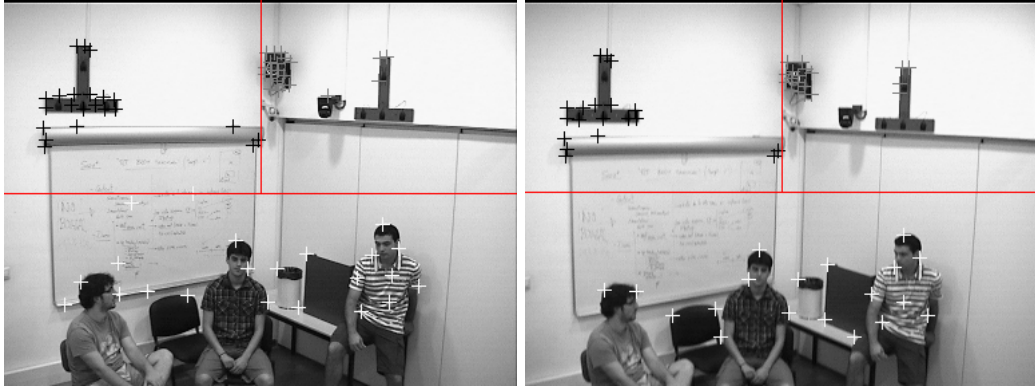


Figura 6.9: Fotogrames amb els punts detectats després del calibratge

Els resultats obtinguts de la compensació de moviment són els que es veuen en la figura 6.10. En aquest cas els resultats no són tan bons com en la compensació amb el mètode de Gauss-Newton degut a l'error d'estimació. De totes formes s'observa com les vibracions han estat compensades i per tant la seqüència de sortida ha quedat estabilitzada.

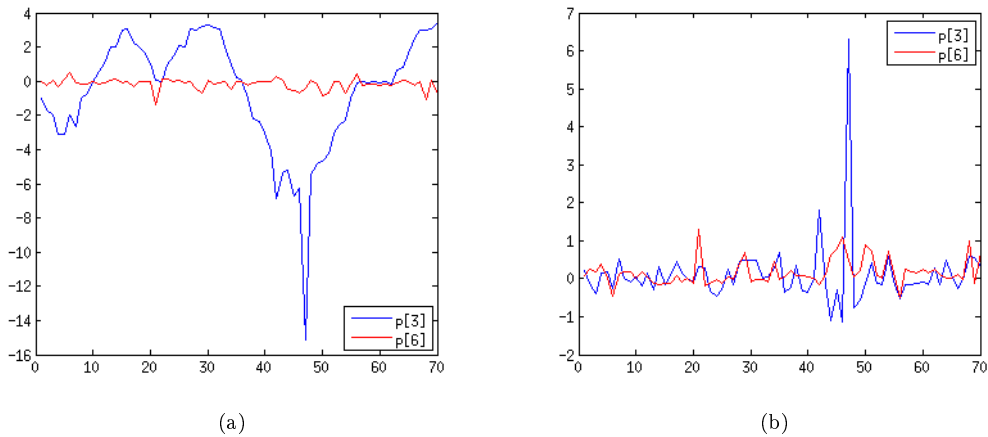


Figura 6.10: Estimació dels paràmetres de moviment de la seqüència estàtica amb el mètode de correlació. En (a) trobem l'estimació de moviment dels paràmetres de desplaçament horitzontal  $p[3]$  i vertical  $p[6]$ . En (b) hi ha l'estimació de moviment sobre la seqüència compensada.

## 6.2 Seqüència amb la càmera amb moviment

Per analitzar aquest cas s'ha creat una seqüència amb una façana d'un edifici d'oficines. En la figura 6.11 podem observar 3 frames d'aquesta seqüència.



Figura 6.11: 3 frames de la seqüència amb moviment

La càmera realitza un moviment oscil·latori conegut. A aquest moviment se li han afegit un parell de vibracions en els eixos transversal i longitudinal. La seqüència s'ha construït a partir d'una imatge per poder disposar dels paràmetres de moviment de la càmera. L'extracció dels paràmetres de moviment es deixa com a treball futur en el capítol 7. Per tant, per poder conèixer els paràmetres de moviment de la càmera hem hagut de crear la seqüència artificialment.

El moviment de la càmera es pot descriure per l'equació recursiva,

$$x(i) = \lfloor x(i-1) + x_c(i) + x_n(i) \rfloor$$

$$y(i) = \lfloor y(i-1) + y_n(i) \rfloor$$

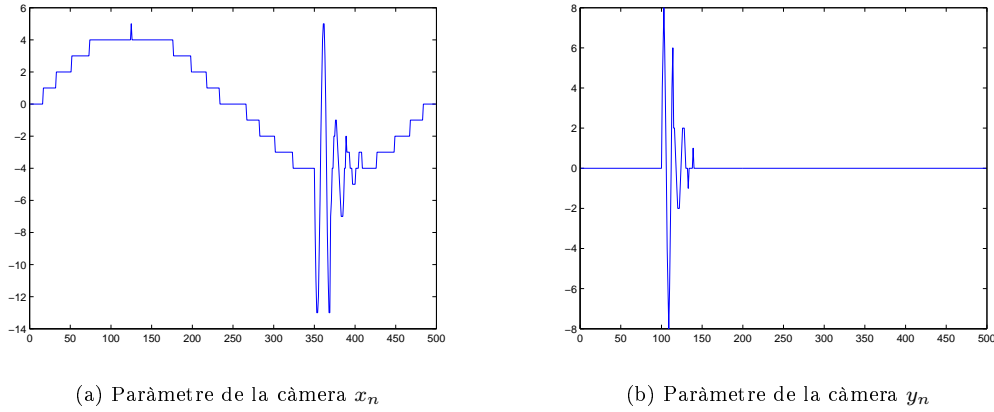
on  $x_c(i)$  representa el moviment de la càmera i ve descrit per

$$x_c(i) = 5 \sin\left(\frac{2\pi}{500}i\right)$$

i  $x_n(i)$  i  $y_n(i)$  representen la vibració en l'eix longitudinal i transversal de la càmera i pretén simular un impuls amb inèrcia. La vibració s'ha simulat,

$$\begin{cases} x_n(i) = 10e^{\frac{350-i}{20}} \sin\left(\frac{2\pi}{15}(i-350)\right) & 350 \leq i \leq 450 \\ x_n(i) = 0 & \text{altrament} \end{cases}$$

$$\begin{cases} y_n(i) = 8e^{\frac{100-i}{15}} \sin\left(\frac{2\pi}{12}(i-100)\right) & 100 \leq i \leq 200 \\ y_n(i) = 0 & \text{altrament} \end{cases}$$

Figura 6.12: Paràmetres de moviment de la càmera  $x_n$  i  $y_n$ 

En la figura 6.12 podem observar una representació de  $x(i)$  i  $y(i)$  que representa el moviment oscil·latori de la càmera amb la vibració.

Per compensar la seqüència de vídeo i eliminar les vibracions haurem d'estimar el moviment i posteriorment calcular els paràmetres per fer la compensació. Els paràmetres per fer la compensació els calcularem com

$$p_c(i) = \hat{p}(i) - p_{ref}(i) + p_c(i-1) \quad (6.1)$$

on  $\hat{p}(i)$  són els paràmetres estimats en el frame  $i$  i  $p_{ref}$  són els paràmetres de la càmera. Recordem que en el cas que la càmera presenti moviment hem d'anar acumulant els paràmetres calculats en la iteració anterior. En aquest cas estem utilitzant el model de transformació afí i per tant la inicialització dels paràmetres de compensació necessaris per calcular  $p_c(1)$  serà la següent,

$$p_c(0) = [100010]$$

els paràmetres  $p_c(0,1)$  i  $p_c(0,4)$  prendran el valor unitat ja que aquests controlen l'escalat de la imatge i en el moment inicial no s'ha de realitzar cap escalat. La resta de paràmetres prendran el valor nul ja que en el principi no s'ha realitzat cap desplaçament ni deformació dels eixos.

La diferència entre la compensació de la seqüència amb el moviment conegut o desconegut radica únicament en l'obtenció dels paràmetres  $p_{ref}$ . En el primer cas els paràmetres els llegirem d'un recurs extern, com per exemple, un fitxer de text. En el segon cas haurem de calcular-los tal com s'ha explicat en la secció 5.3 a la pàgina 44.

L'estimació dels paràmetres de moviment  $\hat{p}$  serà pràcticament igual per als dos casos. L'única diferència la trobem en la inicialització del mètode de descens de gradient. En un cas la inicialització es realitzarà amb els paràmetres de la càmera coneguts i en l'altre amb els paràmetres estimats en la iteració anterior. A continuació analitzarem els resultats obtinguts de l'estimació de moviment amb els dos mètodes esmentats en altres seccions: el mètode de Gauss-Newton i el mètode per correlació.

### Estimació pel mètode de Gauss-Newton

A continuació es presenten els resultats de l'estimació corresponent al mètode de descens de gradient mitjançant l'algorisme de Gauss-Newton. En la figura 6.13 podem veure l'estimació del

paràmetre corresponent al desplaçament de la càmera al llarg de l'eix longitudinal del moviment. Com s'ha descrit anteriorment, en aquest eix la càmera presenta un moviment oscil·latori i té superposada una vibració en forma de soroll impulsional a partir del frame 350. En aquesta figura podem observar l'estimació realitzada mitjançant aquest algoritme i l'error que produeix aquesta estimació ja que disposem dels paràmetres de la càmera per poder calcular-lo. L'error quadràtic mig de l'estimació es calcula segons l'expressió

$$mse = \frac{1}{N} \sum_{i=1}^N (\hat{p}(i) - p_{ref}(i))^2 \quad (6.2)$$

i en aquest cas pren el valor de 0,0954. L'estimació d'aquest paràmetre en concret és bastant bona i l'error de l'estimació és pràcticament despreciable.

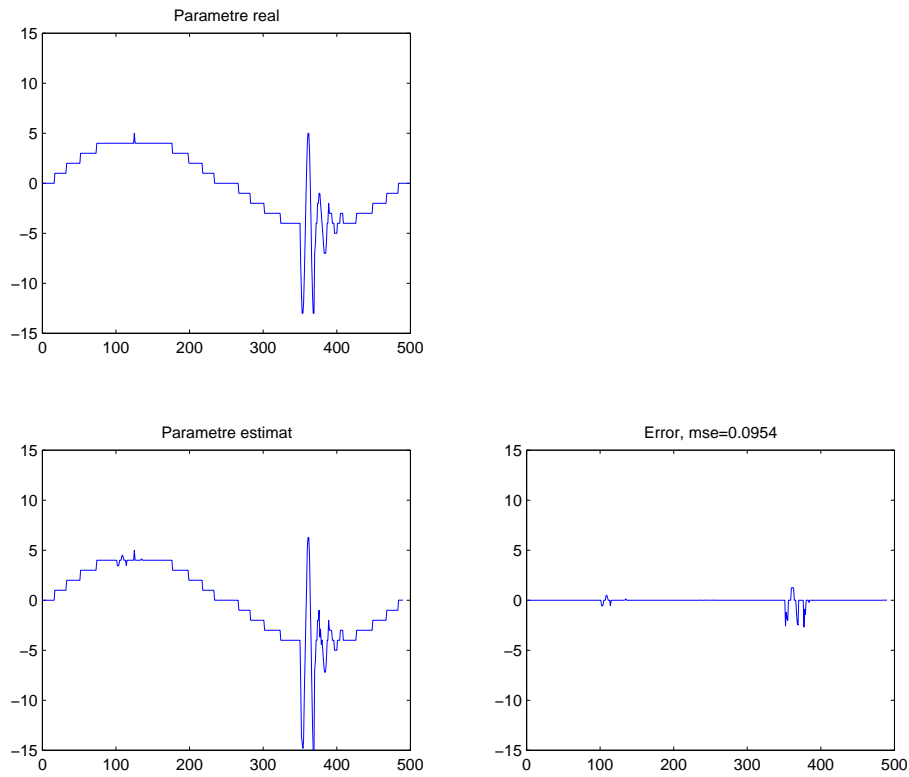
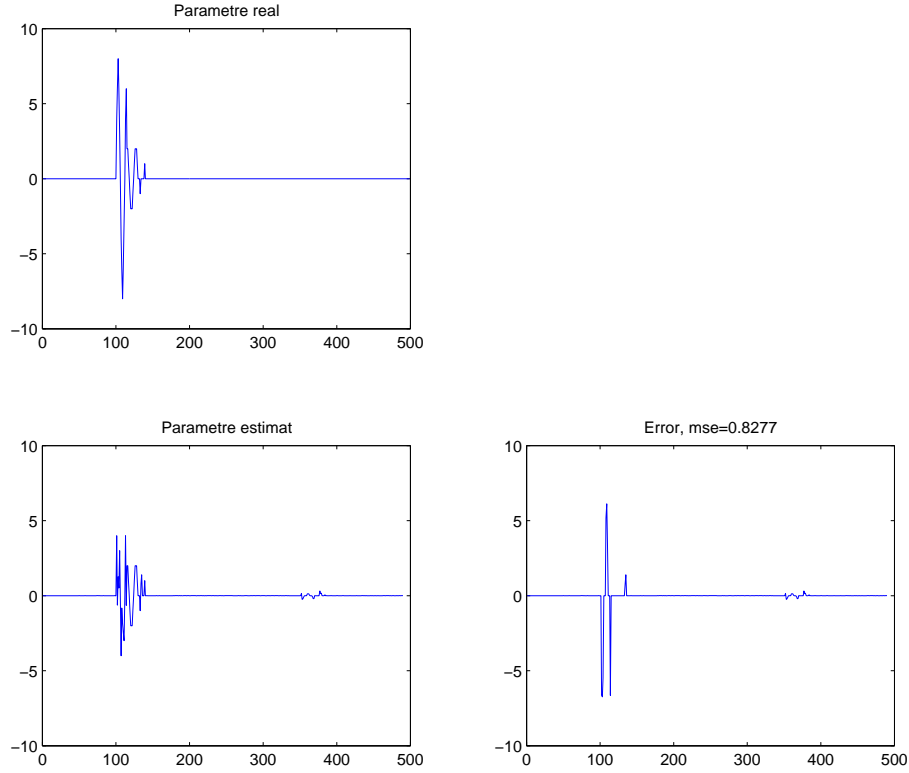


Figura 6.13: Estimació del paràmetre corresponent a  $x_n$

En la figura 6.14 trobem l'estimació del paràmetre corresponent al moviment de la càmera en l'eix transversal al moviment, és a dir,  $y_n$ . Recordem que en aquest eix la càmera només tenia el moviment degut a la vibració que s'ha superposat. En aquest cas l'error d'estimació és superior al del paràmetre anterior sent el seu valor de 0,8277. Aquest error ens podria donar algun problema a l'hora de compensar la seqüència de vídeo. Per intentar millorar-lo podríem ajustar els paràmetres d'aturada (número màxim de passos, i llindars mínims en els increments entre iteracions) del mètode de descens de gradient. Això, però, suposaria una disminució en la velocitat de processament de la seqüència.

Figura 6.14: Estimació del paràmetre corresponent a  $y_n$ 

### Estimació pel mètode de correlació

El primer pas de l'estimació pel mètode de correlació consisteix en la cerca de punts d'interès com s'ha comentat en el capítol 3 a la pàgina 15. Per tal propòsit s'han avaluat les dues modificacions introduïdes en el citat capítol i a continuació es presenten els resultats obtinguts. En la figura 6.15 podem observar els punts detectats segons les dues modificacions explicades en la secció 3.3 a la pàgina 22. Per observar els punts ens hem fixat en un frame de la seqüència de vídeo. Es pot veure que la primera modificació (*detector tres zones*) troba sempre millors punts que la segona (*detector multi-regió*), mentre que el detector multi-regió reparteix els punts uniformement per tota la imatge. Per saber quin d'aquests dos detectors proporciona millors resultats ens haurem de fixar en l'estimació de moviment. Recordem que l'estimació de moviment es fa només amb els punts detectats, per tant, com millors i més ben distribuïts estiguin els punts millor serà l'estimació del moviment de la seqüència.

Per obtenir els punts mitjançant el detector tres zones hem hagut de realitzar un procés de calibratge previ que ens ha proporcionat els següents valors:

$$\begin{cases} \emptyset=5 \\ threshold = 500 \\ d_{min} = 20 \end{cases}$$

Per aquests valors del detector els punts queden més o menys ben distribuïts al llarg de tota la imatge.

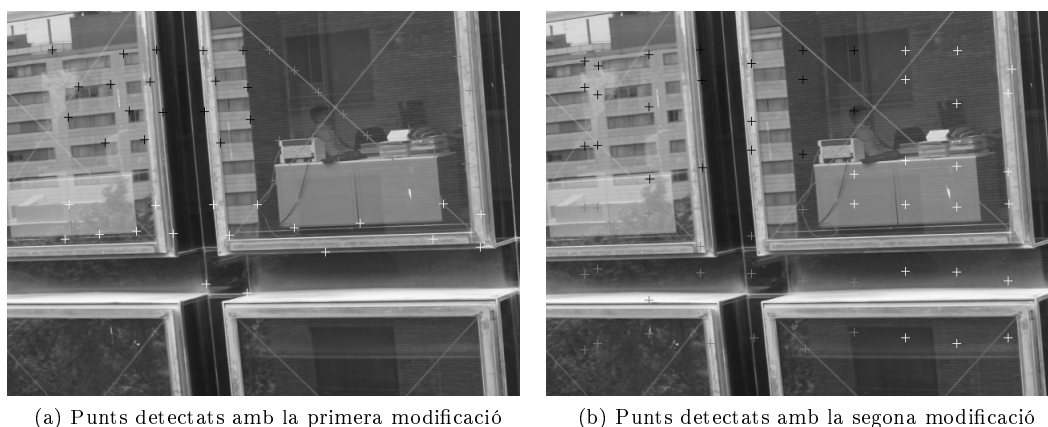


Figura 6.15: Detecció de punts amb el detector de Trajkovic-Headley modificat

En la figura 6.16 trobem l'estimació dels paràmetres amb el detector tres zones. L'estimació dels paràmetres és molt bona ja que l'error quadràtic mig resultant de l'estimació és pràcticament nul. Fixant-nos en les gràfiques d'error podem observar que l'estimació dels paràmetres de moviment és pràcticament idèntica als paràmetres reals de la càmera. A més a més, en la seqüència de sortida es veu com les vibracions s'han compensat totalment.

En la figura 6.17 hi ha representada l'estimació dels paràmetres de moviment utilitzant els punts del detector multi-regió. L'error quadràtic mig que ens dona aquesta estimació és bastant millor al que s'obté amb l'estimador de descens de gradient però per contra és més elevat que l'obtingut amb l'estimador tres zones. A més a més l'error produït per l'estimador multi-regió està més o menys distribuït per tota la seqüència mentre que l'error produït per l'estimador tres zones està bastant concentrat en alguns frames. Això és degut a que la definició del detector multi-regió fa que sempre es trobin el mateix nombre de punts. Si en una determinada regió no hi ha cap punt d'interès, el detector busca el punt amb major nivell de SAD. Això produeix que alguns dels punts no siguin gaire bons per a l'estimació mitjançant la correlació. Per altra banda el detector tres zones només detecta els punts d'interès vertaders. Si en algun fotograma no hi ha gaires punts adequats, llavors l'estimació dels paràmetres no serà tant bona com podem veure en el fotograma 376 de la figura 6.16.

Observant els vídeos de sortida es veu una petita fluctuació en la seqüència obtinguda amb el detector multi-regió. Aquesta fluctuació és inexistent en el vídeo compensat amb els punts obtinguts amb detector tres zones. Així doncs, per aquesta seqüència de vídeo, el detector tres zones ens proporciona millors resultats.

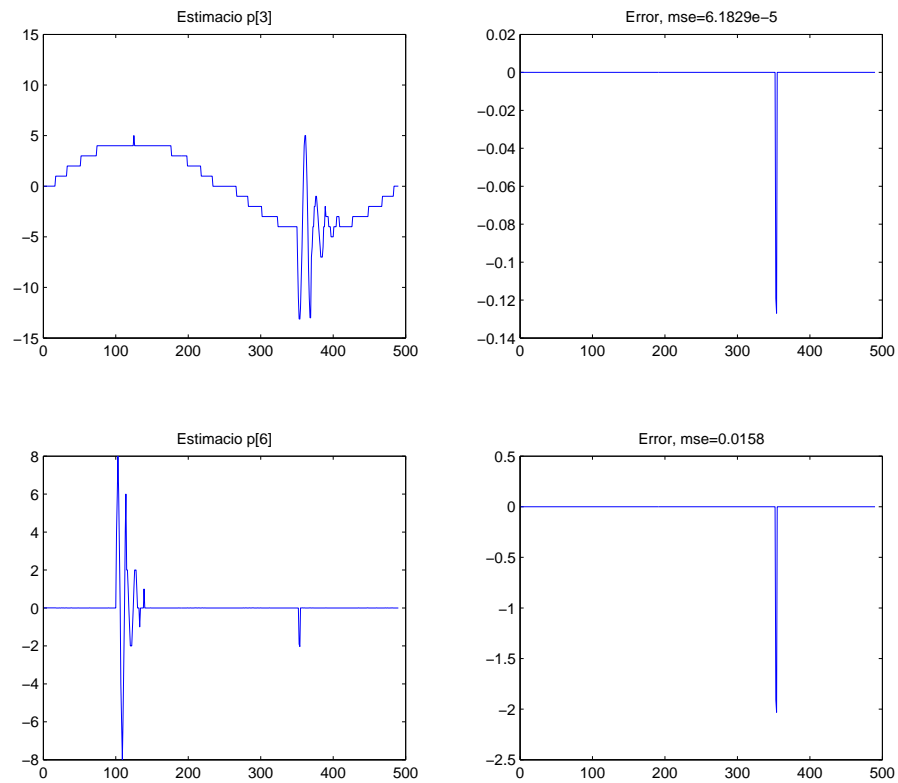


Figura 6.16: Estimació dels paràmetres de moviment amb la primera modificació del detector

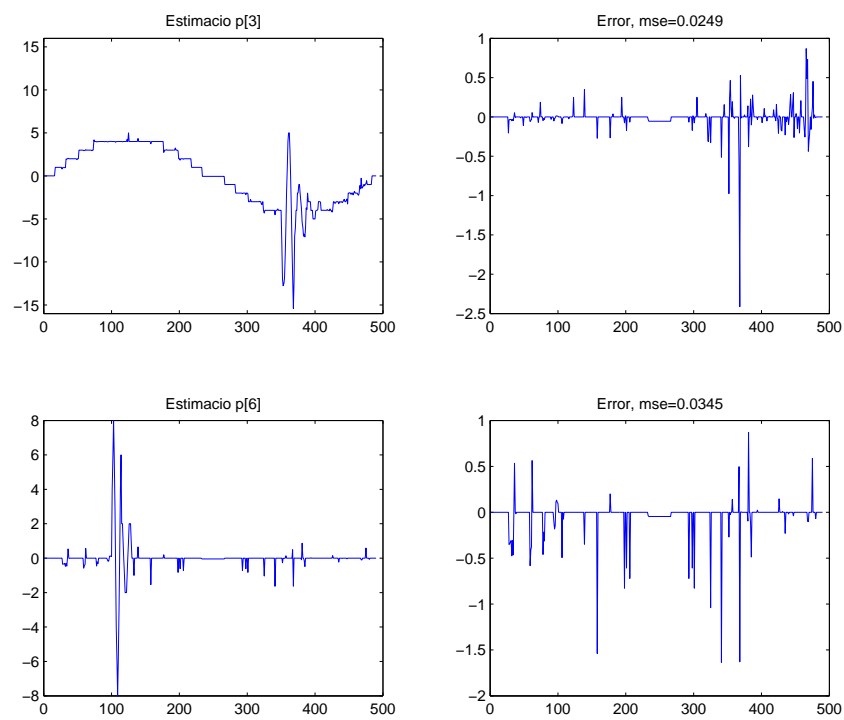


Figura 6.17: Estimació dels paràmetres de moviment amb la segona modificació del detector



Un cop obtinguda l'estimació dels paràmetres de moviment ( $\hat{p}$ ) s'ha de realitzar el càlcul dels paràmetres de compensació ( $p_c$ ) i compensar la seqüència d'entrada per obtenir el vídeo lliure de vibracions. En aquest punt hem de distingir els dos casos que estem estudiant ja que l'obtenció dels paràmetres de referència ( $p_{ref}$ ) és diferent.

### 6.2.1 Compensació amb el moviment conegut

En el primer cas d'estudi els paràmetres de la càmera són coneguts. Per tant, només serà necessari llegir-los d'un fitxer per tenir els paràmetres  $p_{ref}$  i poder obtenir els paràmetres  $p_c$  segons l'equació 6.1. Així doncs per a cada frame que llegim de la seqüència d'entrada estimarem els paràmetres de moviment amb algun dels dos mètodes presentats i posteriorment calcularem els paràmetres de compensació per realitzar-la directament sobre el fotograma que acabem de llegir. L'algoritme es pot revisar a la figura 5.6 a la pàgina 46.

En les figures 6.18, 6.19 i 6.20 es poden observar les representacions dels paràmetres de compensació ( $p_c$ ) calculats a partir de les estimacions dels paràmetres de moviment amb l'estimador de descens de gradient i correlació successivament. Si ens fixem en els paràmetres estimats i la seva compensació podem veure clarament com és va acumulant la compensació entre frames consecutius. On es veu més clarament és en la figura 6.19 en la compensació del paràmetre  $p_c(6)$ . En el fotograma 350 apareix un desplaçament de dos píxels en l'estimació de moviment. Aquests dos píxels es compensen fins al final de la seqüència.

Precisament degut al fet que els paràmetres es van acumulant es molt important realitzar una bona estimació dels paràmetres de moviment. De no ser així, els errors es van acumulant i al final la seqüència de sortida pot quedar totalment distorsionada per aquest fenomen. La figura on es veu més clarament aquest fet és la 6.20 en la compensació del paràmetre  $p_c(6)$ . Després de compensar la vibració hauria d'haver una compensació de 10 píxels constant fins al final de la seqüència. Degut a l'acumulació d'errors aquesta compensació es va reduint progressivament fins que la diferència arriba a ser bastant notable.

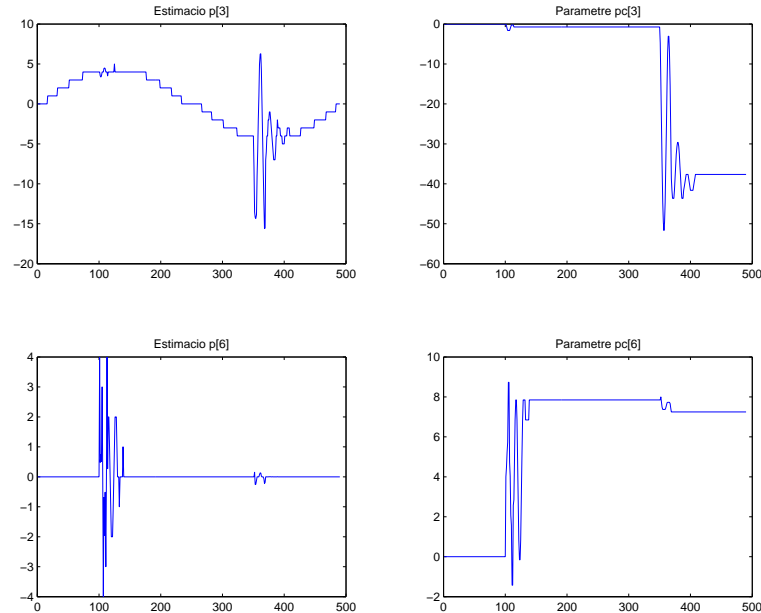


Figura 6.18: Paràmetres calculats a partir de l'estimador de descens de gradient

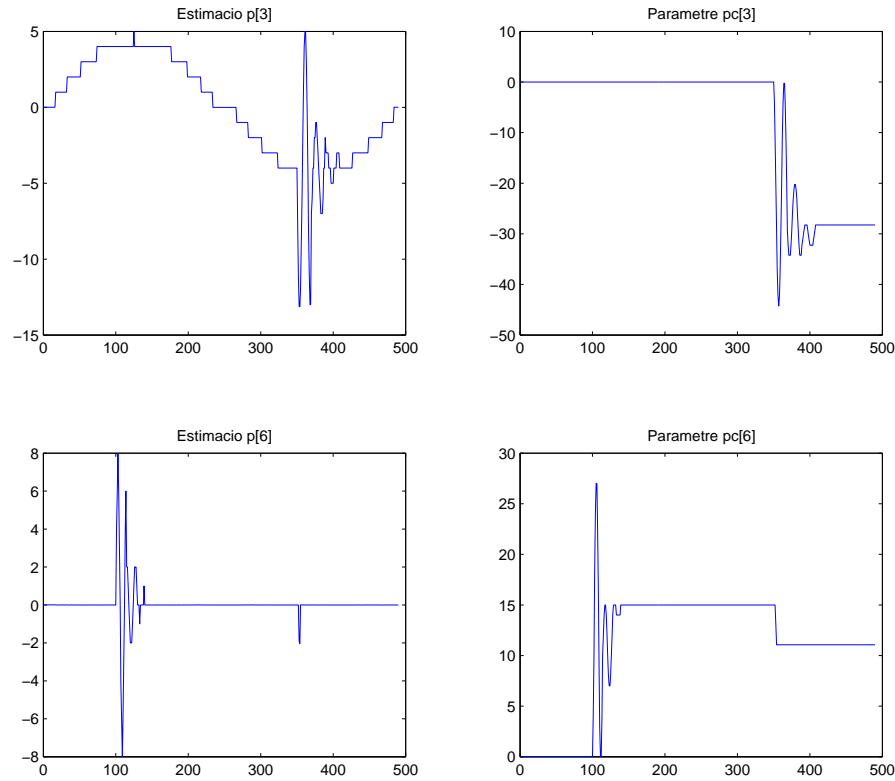


Figura 6.19: Paràmetres calculats a partir de l'estimador de correlació amb els punts del detector 3 zones

### 6.2.2 Compensació amb moviment desconegut

Per tal de poder avaluar els resultats utilitzarem la mateixa seqüència que en el cas anterior. La diferència és que en aquest cas no utilitzarem els paràmetres de la càmera sinó que farem com si aquests fossin desconeguts. Un cop calculats els paràmetres de referència ( $p_{ref}$ ) la compensació es realitza de la mateixa forma que en el cas anterior. Per calcular els paràmetres de moviment de la càmera lliures de vibracions només ho podem fer a partir dels paràmetres estimats. Com la vibració i el moviment de la càmera estan en bandes freqüencials diferenciades podem realitzar un filtrat de l'estimació per tal d'aproximar els paràmetres de la càmera. En la figura 6.21 trobem els paràmetres calculats a partir de les diferents estimacions dels paràmetres de moviment. Els resultats obtinguts per al càlcul dels paràmetres de referència són molt bons. L'error entre els paràmetres calculats i els reals és bastant petit i en cap cas ha estat superior a un píxel.

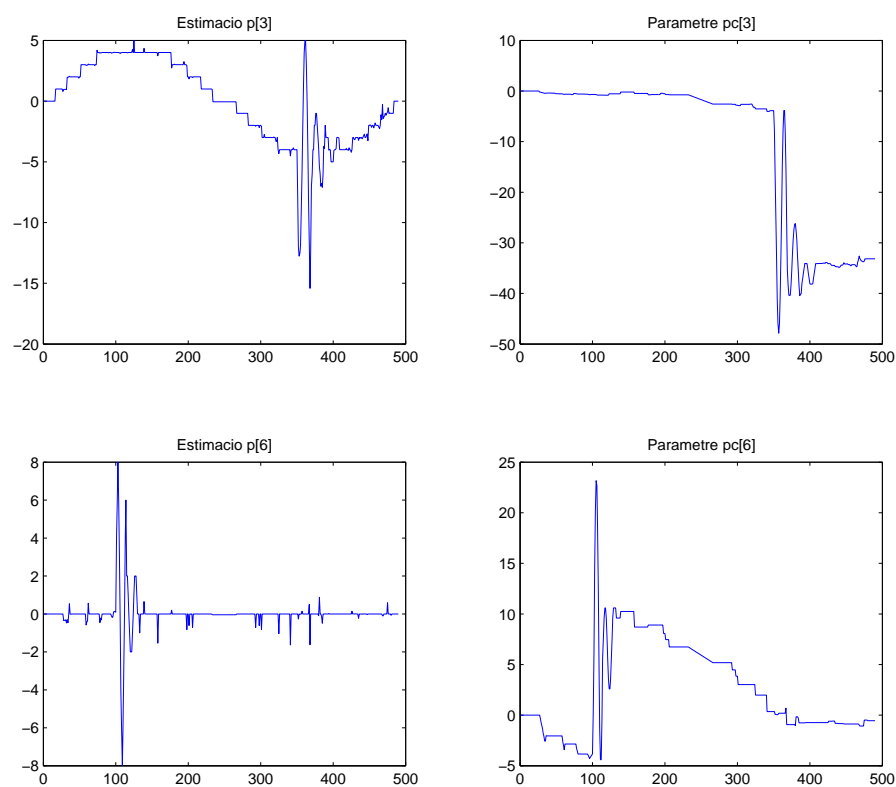
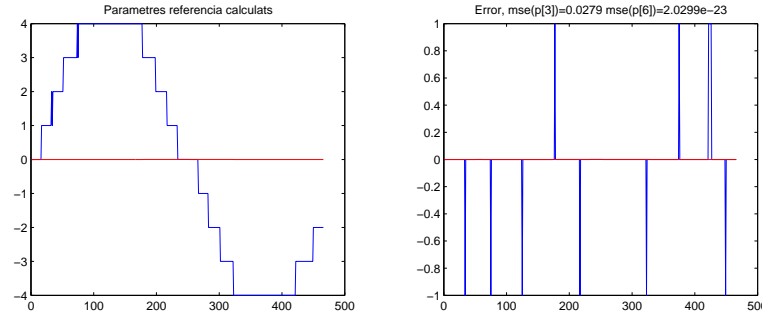
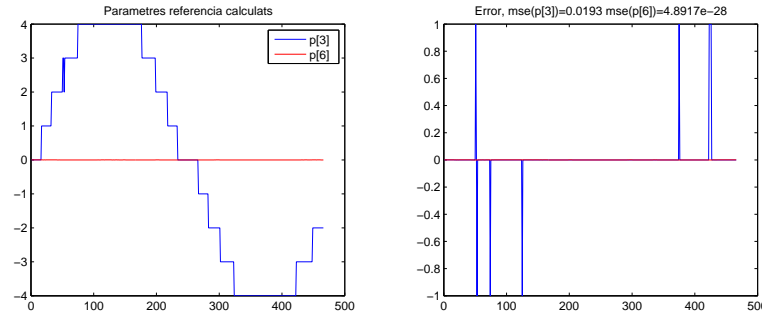


Figura 6.20: Paràmetres calculats a partir de l'estimador de correlació amb els punts del detector multi-regió

En el punt anterior s'ha vist que l'estimació pel mètode de correlació dóna millors resultats amb el detector tres zones. Per tant, per calcular els paràmetres de referència només s'ha tingut en compte aquest detector per a l'estimació pel mètode de correlació.



(a) Paràmetres referència calculats a partir de l'estimació de moviment pel mètode de descens de gradient



(b) Paràmetres de referència calculats a partir de l'estimació de moviment pel mètode de correlació. El detector utilitzat ha estat el tres zones.

Figura 6.21: Paràmetres  $p_{ref}$  calculats. A l'esquerra el paràmetre calculat i a la dreta l'error entre el paràmetre calculat i el paràmetre de la càmera real.

En la figura 6.22 es pot observar la diferència entre els paràmetres de compensació calculats en aquest apartat i els obtinguts en el punt anterior. Recordem que els paràmetres de compensació del punt anterior es calculaven amb els paràmetres reals de la càmera, mentre que en aquest apartat els hem calculat a partir de l'estimació de moviment. La diferència entre uns i altres està bastant acotada i si ens fixem en el cas dels paràmetres obtinguts per l'estimador de correlació observarem que la diferència és bastant menor. Això és degut a que l'estimació amb aquest mètode era millor, tal com hem vist en el punt 6.2.1.

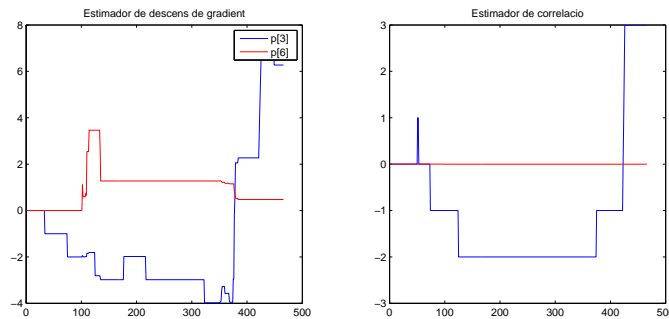


Figura 6.22: Diferència entre els paràmetres de compensació calculats a partir dels paràmetres reals de la càmera i els paràmetres de referència calculats a partir de l'estimació

Igual que en el cas de la seqüència estàtica, s'ha enregistrat una escena en el laboratori del TSC. En l'escena la càmera descriu un moviment horitzontal al que se li ha afegit soroll en forma de vibracions. En la figura 6.23 es poden observar alguns dels fotogrames d'aquesta seqüència.

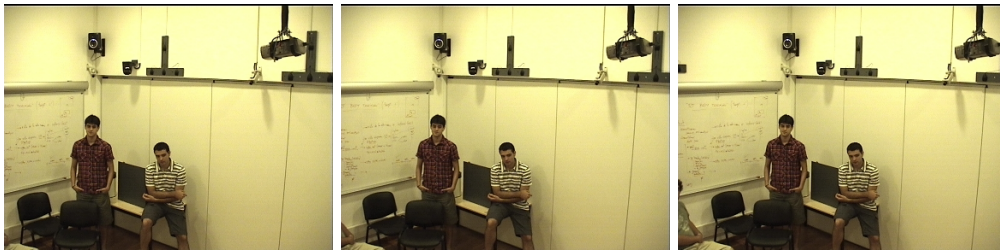
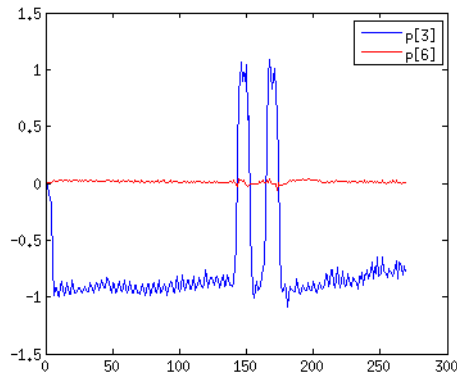
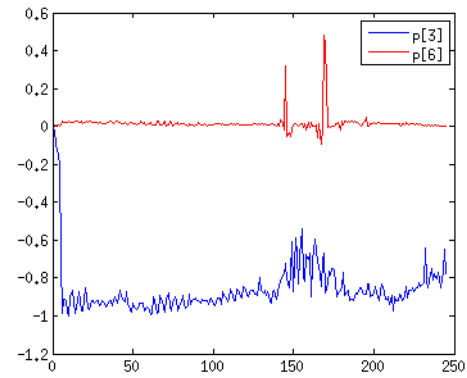


Figura 6.23: 3 fotogrames de la seqüència del laboratori amb la càmera amb moviment.

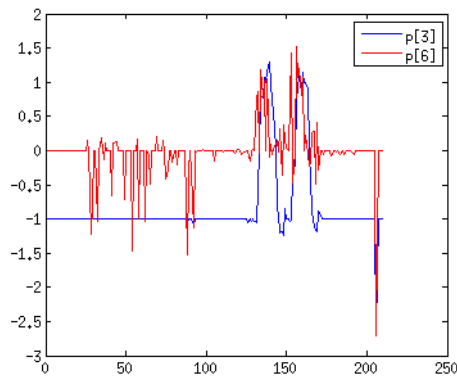
En la figura 6.24 es poden observar els resultats obtinguts de la compensació de vibracions per mitjà dels dos mètodes d'estimació de moviment desenvolupats. L'offset de -1 píxel en el paràmetre  $p[3]$  indica que la càmera realitza un moviment horitzontal constant. Al voltant del fotograma 150 apareix una vibració indesitjada sobre la càmera. Es pot observar com el mètode de descens de gradient, en aquest cas, realitza una estimació més acurada dels paràmetres de moviment. La compensació és gairebé perfecta excepte el soroll residual que apareix en el moment de la vibració. En l'estimació per correlació la compensació de la vibració també es produeix correctament. Malgrat això l'error d'estimació introduït per aquest estimador, paràmetre  $p[6]$  en la figura 6.24c, afecta el vídeo de sortida introduint petites fluctuacions com es veu en la figura 6.24d.



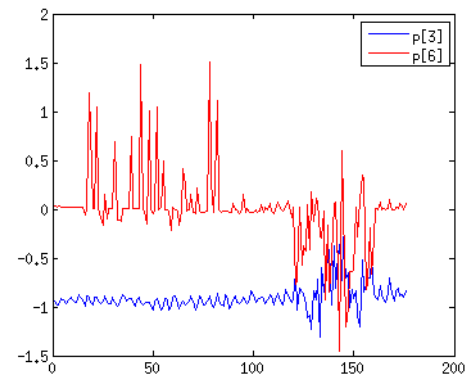
(a) Estimació pel mètode GNA



(b) Paràmetres després de la compensació per GNA



(c) Estimació pel mètode de correlació



(d) Paràmetres després de la compensació per correlació

Figura 6.24: En 6.24a i 6.24b trobem l'estimació dels paràmetres  $p[3]$  i  $p[6]$  mitjançant el mètode de descens de gradient. En 6.24c i 6.24d hi ha l'estimació dels paràmetres mitjançant el mètode de correlació. En els dos casos s'observa en primer lloc l'estimació dels paràmetres sobre el vídeo original i, posteriorment, l'estimació dels paràmetres sobre el vídeo compensat.

## Capítol 7

# Conclusions i treball futur

### 7.1 Conclusions

Per tal de millorar la qualitat de les seqüències de vídeo enregistrades per sistemes de videovigilància s'ha desenvolupat un software capaç d'estabilitzar-les. Aquest software processa les seqüències estimant el moviment i compensant les vibracions indesitjades. Per una banda, la millora de la qualitat dels enregistraments ajuda als operaris que revisen els monitors. Les vibracions en les seqüències de vídeo són molt molestes i més si s'han de vigilar durant llargs períodes de temps. Per altra banda, hi ha sistemes de processament d'imatge com ara reconeixement de cares, seguiment d'objectes, lectura de matrícules, etc., que requereixen seqüències estabilitzades per poder treballar de forma correcta. Un altre avantatge és que les seqüències de vídeo estabilitzades contenen menys quantitat d'informació i es poden comprimir més a l'hora de ser emmagatzemades.

A l'inici d'aquest projecte plantejàvem dos objectius. El primer d'ells consistia en la creació d'un software capaç de compensar les vibracions en una seqüència de vídeo capturada per una càmera de videovigilància. Al llarg d'aquest treball s'han anat presentant diverses formes de realitzar aquesta compensació. També s'han presentat diverses situacions on es podria aplicar la compensació de vibracions. Segons la naturalesa de la seqüència de vídeo s'ha distingit entre l'eliminació de vibracions per a seqüències estàtiques, seqüències on coneixíem els paràmetres de moviment de la càmera i seqüències amb moviment de la càmera desconegut a priori. En els resultats s'ha pogut observar com es compensaven les seqüències de vídeo en cada cas.

El segon objectiu del projecte consistia en augmentar la velocitat de processament fins a poder compensar les seqüències de vídeo en temps real, entre 12 i 25 fotogrames per segon. Aquest objectiu s'havia fixat per poder arribar a produir un sistema comercial capaç d'enregistrar les seqüències ja lliures de vibracions, és a dir, processant-les abans d'enregistrar-les en els sistemes d'emmagatzematge o abans de presentar-les en els monitors de vigilància. En primer lloc s'ha avaluat l'estimador de Gauss-Newton. Aquest estimador, però, requereix un gran cost computacional. La velocitat de processament de les seqüències de vídeo amb aquest estimador està al voltant de 0,5fps. Per augmentar la velocitat de processament s'ha desenvolupat un altre estimador basat en la correlació. La correlació entre imatges s'ha implementat mitjançant la FFT per intentar aconseguir reduir la càrrega computacional. La velocitat de processament amb aquest estimador s'ha augmentat uns 4 cops fins a obtenir velocitats de processament d'uns 2fps. Aquest augment no és suficient per poder processar les imatges en temps real. Malgrat això, els resultats obtinguts amb aquest estimador han estat força bons. Així doncs, tot i que s'ha obtingut una millora en el temps de processament de les seqüències de vídeo no ha estat possible assolir l'objectiu plantejat. De totes formes, cal destacar que l'objectiu era molt ambiciós ja que es pretenia augmentar com a mínim 25 cops la velocitat de processament.

A més a més, degut al procés de miniaturització del transistor que segons la llei de Moore es duplica el nombre de transistors per unitat de superfície cada 18 mesos, s'arriba a la conclusió que l'aplicació podria funcionar a temps real en uns dos anys i mig aproximadament.

Com s'ha comentat durant el projecte, un dels punts claus en la compensació de vibracions consisteix en l'estimació de moviment. Per tal d'estimar els paràmetres de moviment de les imatges s'han presentat dos estimadors. El primer d'ells es basa en el mètode de Gauss-Newton. Aquest és un mètode iteratiu per descens de gradient. L'estimador de Gauss-Newton s'ha modificat seguint un mètode multi-nivell. Amb això s'aconsegueix evitar que l'estimador caigui en mínims locals i també eliminar la dependència dels resultats de l'estimador amb la seva inicialització. La convergència i la velocitat de convergència del mètode de Gauss-Newton depèn en gran mesura de la inicialització del model. Amb el mètode multi-nivell s'aconsegueix que aquesta dependència sigui menys forta. Per tal d'aconseguir-ho s'ha d'inicialitzar cadascun dels nivells successius amb el resultat obtingut en el nivell anterior. Aquest mètode també s'anomena 'coarse to fine', és a dir, en primer lloc es busca una aproximació del resultat i posteriorment es va refinant el resultat amb els següents nivells.

El segon estimador que s'ha estudiat és el basat en la correlació. El desenvolupament d'aquest estimador ha portat a l'estudi de la modificació d'un detector de punts d'interès. Durant l'execució del treball s'ha modificat el detector de Trajkovic-Headley. Aquest detector cerca els punts d'interès d'una imatge buscant els píxels que maximitzen una funció SAD (Sum of Absolute Differences). Aquesta funció calcula la diferència absoluta entre els píxels que hi ha al voltant d'un punt i donat un llindar estableix com a punts d'interès aquells que presenten un valor per al SAD superior a aquest. L'estimació dels paràmetres de moviment, a més de buscar els punts d'interès d'una imatge requereix que aquests estiguin uniformement distribuïts per tota la imatge. A més a més, per estimar els paràmetres de moviment s'han d'agafar els punts detectats de 3 en 3 per resoldre un sistema d'equacions. Per poder resoldre aquest sistema els punts no poden estar alineats ja que si ho estigueren, les equacions del sistema a resoldre serien linealment dependents i el sistema no es podria resoldre. Amb aquestes dues premisses s'han realitzat dues modificacions en el detector.

La primera d'elles, que hem anomenat detector de 3 zones simplement consistia en dividir la imatge en 3 zones i aplicar el detector original en cadascuna d'aquestes zones. Gràcies a aquesta modificació els punts que agafem per resoldre el model de transformació mai estan alineats. La segona modificació que s'ha realitzat ha estat anomenada detector multi-regió i divideix la imatge en tantes regions com punts volem trobar. Per cada regió busca el punt amb màxima SAD.

La modificació anomenada detector multi-regió presenta l'avantatge de distribuir uniformement els punts per tota la imatge. El principal inconvenient és que si les imatges que conformen la seqüència tenen zones amb poca textura, els punts detectats en aquestes regions no són gaire bons per a l'estimació mitjançant correlació. Així doncs, el que a priori semblava una millora per al detector de punts d'interès ha resultat no ser tan bona degut a la naturalesa de les seqüències utilitzades. De totes formes, els resultats obtinguts amb tots dos detectors han estat satisfactoris per a l'aplicació desenvolupada com s'ha pogut observar en el capítol de resultats.

Durant l'execució d'aquest treball s'ha pogut observar que el resultat de la compensació de vibracions depèn en gran mesura de les imatges captades en la seqüència de vídeo. Com en totes les aplicacions de processament d'imatge els resultats són bastant sensibles a situacions d'il·luminació desfavorables, com podrien ser la manca de lluminositat de les imatges o canvis bruscos en aquesta. Una altra situació desfavorable que ens hem trobat ha estat la manca de textura en les imatges d'una seqüència de vídeo. És a dir, zones de la imatge sense punts d'interès per poder processar la imatge correctament.

Un altra de les conclusions que es poden extreure del treball realitzat és que els resultats de la compensació de vibracions estan totalment lligats a l'estimació dels paràmetres de moviment. Per tant, el punt més crític de l'aplicació és la forma en com es calculen aquests paràmetres.



Una mala estimació pot produir, fins i tot, més vibracions en la seqüència de sortida de les que existien en el vídeo original degut al soroll d'estimació. D'aquesta forma, s'ha vist que no hi ha un mètode millor per a qualsevol cas sinó que depenent de les imatges funciona millor un estimador que un altre. És per això que per compensar les vibracions d'una determinada seqüència de vídeo en primer lloc s'ha de buscar el millor estimador per a la seqüència en concret. Això s'hauria de fer en una fase de calibratge i prenent algunes seqüències de la zona de treball de la càmera.

Finalment cal remarcar que en tots els casos en que s'ha estudiat la compensació de vibracions s'han obtingut uns resultats favorables, podent compensar en cada cas les seqüències de vídeo i eliminant així les vibracions indesitjades.

Durant el desenvolupament del projecte s'ha realitzat la programació en C d'algunes funcions i programes dintre de les llibreries de Softimage del TSC. Els programes que s'han desenvolupat són:

- Programa de compensació de seqüències. És el programa principal del projecte i respon al diagrama 5.6 a la pàgina 46. Aquest programa llegeix un fitxer de configuració així com la seqüència de vídeo, estima el moviment, realitza l'estabilització i emmagatzema la seqüència resultant.
- Programa per a la creació de seqüències artificials. Aquest programa s'ha utilitzat per generar les seqüències de prova i poder disposar dels paràmetres de moviment d'aquesta.

També s'han desenvolupat les següents funcions que utilitza el programa de compensació de seqüències:

- Estimador per correlació. S'ha elaborat una funció que estima els paràmetres de moviment mitjançant la correlació. Aquesta funció utilitza els detectors de punts d'interès per buscar la correlació només d'aquells punts on hi ha major quantitat d'informació dintre del frame que s'està processant. La correlació de les imatges s'ha implementat mitjançant la transformada ràpida de Fourier o FFT.
- Detectors de punts d'interès. S'han programat dos detectors de punts d'interès, cadascun corresponent a una modificació del detector de Trajkovic-Headley.
- Filtrat de mediana. Aquesta funció s'utilitza per obtenir els paràmetres de referència quan es processen les seqüències en moviment sense conèixer prèviament els paràmetres de la càmera. Els paràmetres de referència es calculen amb el filtrat de mediana dels paràmetres estimats.
- Funcions per treballar amb matrius. Per tal de realitzar el filtrat dels paràmetres estimats es necessari emmagatzemar tants paràmetres com l'ordre del filtre. Aquests paràmetres s'emmagatzemen en forma matricial, cada fila de la matriu correspon a un vector de paràmetres de moviment. S'han desenvolupat les funcions per inserir i extreure vectors de les matrius de forma ràpida i senzilla.

El programa de compensació de vibracions ens dona els paràmetres estimats, els paràmetres de referència i els paràmetres de compensació en fitxers de text. Tots els gràfics presentats en els resultats corresponen a representacions d'aquests paràmetres i s'han realitzat mitjançant Matlab.

## 7.2 Treball futur

Durant el desenvolupament de l'aplicació han anat sortit punts que no s'han tractat ja que no entraven en l'àmbit inicial del projecte. Aquestes punts, però, podrien suposar grans millores

per a l'aplicació i es per això que es proposen en aquesta secció. Aquí no es pretén donar cap pauta d'investigació sinó que simplement es tracta de proposar alguns temes que podrien ajudar a millorar el treball realitzat.

Així doncs podem distingir dos línies clarament diferenciades. La primera d'elles consisteix en el càlcul dels paràmetres de la càmera. Com s'ha vist en els resultats l'aplicació funciona lleugerament millor quan els paràmetres de la càmera són coneguts. Per tant, una forma fàcil de millorar l'aplicació seria buscant per a totes les seqüències a tractar els paràmetres de moviment de la càmera.

Sovint, les càmeres de videovigilància amb moviment, són controlades per un servomotor. Per conèixer els paràmetres de moviment de la càmera seria necessari conèixer en primer lloc el moviment que descriu el motor. Això es possible gràcies al programa de control d'aquest. Un cop conegut el moviment que realitza el motor s'hauria d'aplicar una transformació adient per tal de convertir aquest moviment en els paràmetres de moviment de la càmera. Suposadament el moviment de la càmera es realitza amb un motor pas a pas controlat per un programa. El programa diu al motor quants passos ha de realitzar. Així doncs un procés de calibratge podria consistir en posar una pantalla just al davant de la zona que es vol controlar amb la càmera. En aquesta pantalla es dibuixa una forma coneguda, tal com es pot veure en la figura 7.1. Un cop fet això es fa moure la càmera un pas de motor i es calculen els paràmetres de moviment entre els dos frames capturats. D'aquesta forma s'aconsegueix la correspondència entre un pas del motor i els paràmetres de moviment corresponents.

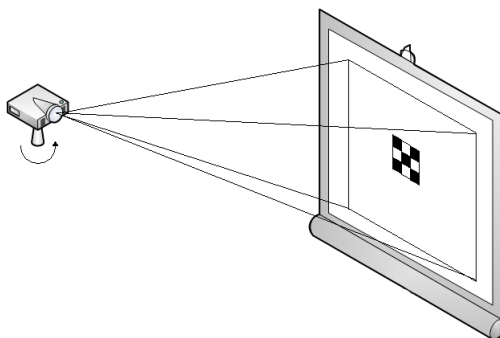


Figura 7.1: Esquema del procés de calibratge

Una altre dels punts que s'ha detectat i que no s'ha pogut tractar durant l'execució del projecte està en l'estimació de moviment. L'estimació de moviment pel mètode de correlació desenvolupat durant aquest treball realitza l'estimació agafant punts de tota la imatge. En el cas que apareguin objectes en moviment al davant de la càmera, aquests no s'haurien de tenir en compte a l'hora de fer l'estimació. Per tant, una de les millores a introduir consisteix en la segmentació de les imatges de la seqüència d'entrada segons el moviment que presenten. S'hauria de poder distingir, d'aquesta manera, el moviment principal de la imatge causat pel moviment de la pròpia càmera i el moviment dels objectes que hi ha en l'escena. Un cop segmentades les imatges, els punts corresponents al moviment secundari no s'haurien de tenir en compte a l'hora de realitzar l'estimació de moviment. Amb això s'aconseguirien uns resultats millors, sobretot a les seqüències on apareixen subjectes en moviment.

Finalment un altra de les millores que s'ha detectat i que es podria dur a terme consisteix en la reconstrucció dels fotogrames un cop compensada la seqüència de vídeo. Quan compensem un fotograma apareix una regió en que la imatge resultant de la compensació no es pot reconstruir amb el frame que s'està compensant. Això és degut a que al realitzar la compensació alguns

dels punts necessaris per reconstruir la imatge es troben fora del fotograma original. En la figura 7.2 es pot observa aquest efecte. Una possible solució a aquest efecte seria estudiar el desplaçament màxim que sofreix la imatge i reduir el tamany de la seqüència de sortida. Amb això, però, perdríem una àrea en els vídeos capturats. Un altra solució, millor que l'anterior, consisteix en reconstruir les zones afectades mitjançant tècniques de mosaicing utilitzant els frames precedents o posteriors de la seqüència de vídeo. És a dir, totes les zones de la imatge que no podem trobar en el fotograma original s'haurien de buscar en altres fotogrames. D'aquesta manera es pot aconseguir restaurar tota la imatge i obtenir així un vídeo compensat del mateix tamany que el vídeo original.

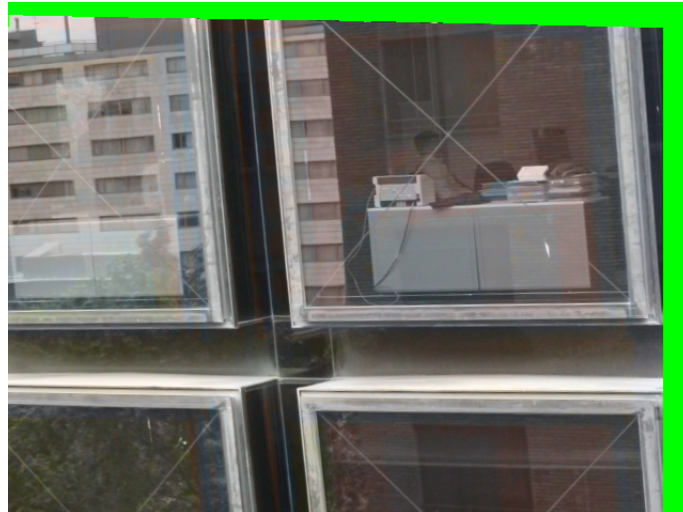


Figura 7.2: Fotograma de la seqüència compensada. En verd s'ha pintat la zona on no s'ha pogut re-dibuixar el fotograma degut a que no es disposava de la informació en el fotograma original.

# Bibliografia

- [Osh89] M. Oshima, T. Hayashi, S. Fujioka, T. Inaji, H. Mitani, J. Kajino, K. Ikeda, k. Komoda. *VHS camcorder with electronic image stabilizer*. IEEE Transactions on Consumer Electronics, Vol. 35, Issue 4, Pàgines: 749 - 758. Novembre 1989.
- [Sato93] K. Sato, S. Ishizuka, A. Nikami and M. Sato. *Control techniques for optical image stabilizing system*. IEEE Transaction on Consumer Electronics, Vol. 39, Issue 3, Pàgines: 461 - 466. Agost 1993.
- [Canon10] <http://www.canon.com/bctv/faq/vari.html>
- [Yong09] Zhang Yong-xiang, Zhang Wei-gong, Zaho Xiao-xu and Yuan Hui-mei. *Study on Electronic Image Stabilization System Based on MEMS Gyro*. International Conference on Electronic Computer Technology. Pàgines 641 - 643. Febrer 2009.
- [Shi07] P. Shi, Y. Zhu and S. Tong. *Video Stabilization in Visual Prosthetics*. IEEE International Conference on Complex Medical Engineering. Pàgines 782-785. Maig 2007.
- [Kwon07] Ohyun Kwon, Byungdeok Nam. *Motion Vector Analysis Algorithm for Selective and Robust Digital Image Stabilization*. International Conference on Consumer Electronics. Digest of Technical Papers. Pàgines 1-2. Gener 2007
- [Hu07] R. Hu, R. Shi, I. Shen, W. Chen. *Video Stabilization Using Scale-Invariant Features*. 11th International Conference on information Visualization. Pàgines 871-877. Juliol 2007.
- [Yuan07] Fei Yuan, Hong Zhang, Ruiming Jia. *Digital Image Stabilization Based on Log-Polar Transform*. Fourth International Conference on Image and Graphics. Pàgines 769-773. Agost 2007.
- [Nait07] A. Naït-Ali. *Genetic Algorithms for Blind Digital Image Stabilization Under Very Low SNR*. IEEE Transactions on Consumer Electronics. Vol. 53, Issue:3. Pàgines 857-863. Agost 2007.
- [Bosco08] A. Bosco, A. Bruna, S. Battiato, G. Bella, G. Puglisi. *Digital video stabilization through curve warping techniques*. IEEE Transactions on Consumer Electronics. Vol. 54, Issue:2. Pàgines 220-224. Maig 2008.
- [Ondrej08] M. Ondrej, Z. Frantisek, D. Martin. *Software video stabilization in a fixed point arithmetic*. First International Conference on the Applications of Digital Information and Web Technologies. Pàgines 389-393. Octubre 2008.
- [Bat08] S. Battiato, G. Puglisi, A. Bruna. *A robust video stabilization system by adaptive motion vectors filtering*. IEEE International Conference on Multimedia and Expo. Pàgines 373-376. Abril 2008.

- [Chao08] Chao-Ho Chen, Che-Wei Hsu, Chin-Hsing Chen. *Video Stabilization by Feature-Block Motion Estimation*. Eighth International Conference on Intelligent Systems Design and Applications. Vol. 3. Pàgines 557-560. Desembre 2008.
- [Yong10] Zhang Yong-xiang, Wang Jian, Zhang Wei-gong, Zhou Quan, Ding Rui. *An improved algorithm of electronic image stability based on block matching*. 2010 the 5th IEEE Conference on Industrial Electronics and Applications. Pàgines 1924-1927. Juliol 2010.
- [Harris98] C. Harris and M. Stephens. *A combined corner and edge detector*. Proceedings of the 4th Alvey Vision Conference. Pàgines 147 - 151. 1998.
- [Shi94] J. Shi and C. Tomasi. *Good Features to Track*. 9th IEEE Conference on Computer Vision and Pattern Recognition. June 1994.
- [Tra98] M. Trajkovic and M. Hedley. *Fast corner detection*. Image and Vision Computing 16, pàgines 75-87. 1998.
- [Smith97] S. M. Smith and J. M. Brady. *SUSAN - a new approach to low level image processing*. International Journal of Computer Vision 23, pàgines 45-78. Maig de 1997.
- [Tene00] J. B. Tenenbaum, V. de Silva, J. C. Langford (2000). *A global geometric framework for nonlinear dimensionality reduction*. Science 290 (5500): 2319-2323, 22 December 2000.
- [fftw] <http://www.fftw.org>
- [Werner07] Werner Van Belle. *An Adaptive Filter for the Correct Localization of Subimages: FFT based Subimage Localization Requires Image Normalization to work properly*. Signal Processing; Yellowcouch. Octubre 2007